Software Engineering Institute

Introduction to the Security Engineering Risk Analysis (SERA) Framework

Christopher Alberts
Carol Woody
Audrey Dorofee

November 2014

TECHNICAL NOTE
CMU/SEI-2014-TN-025

CERT Division

http://www.sei.cmu.edu

Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

This report was prepared for the
SEI Administrative Agent
AFLCMC/PZM
20 Schilling Circle, Bldg. 1305, 3rd floor
Hanscom AFB, MA 01731-2125

DM-0001881

Table of Contents

List of Figures

List of Tables

Acknowledgments

This report describes our initial phase of research into early lifecycle risk analysis. We would like to thank Kevin Fall, Chief Technical Officer of the Software Engineering Institute, for providing the initial research funding for this work. We would like to thank those members of the software acquisition-and-development community who reviewed our early conceptual designs and provided our initial pilot opportunities. We were fortunate to have support from both the Department of Defense (DoD) and Federal Civilian Agencies. We would also like to thank the participants at our challenge problem workshop in August 2014. Their thoughtful feedback helped us to improve our approach and enabled us to chart a course for future research and development related to this work. Finally, we would like to thank Rita Creel and Stephen Blanchette for reviewing this report.

Abstract

Software is a growing component of modern business- and mission-critical systems. As organizations become more dependent on software, security-related risks to their organizational missions are also increasing. Traditional security-engineering approaches rely on addressing security risks during the operation and maintenance of software-reliant systems. However, the costs required to control security risks increase significantly when organizations wait until systems are deployed to address those risks. It is more cost effective to address software security risks as early in the lifecycle as possible. As a result, researchers from the CERT® Division of the Software Engineering Institute (SEI) have started investigating early lifecycle security risk analysis (i.e., during requirements, architecture, and design). This report introduces the Security Engineering Risk Analysis (SERA) Framework, a model-based approach for analyzing complex security risks in software-reliant systems and systems of systems early in the lifecycle. The framework integrates system and software engineering with operational security by requiring engineers to analyze operational security risks as software-reliant systems are acquired and developed. Initial research activities have focused on specifying security requirements for these systems. This report describes the SERA Framework and provides examples of pilot results.

1 Introduction

Software is a growing component of modern business- and mission-critical systems. As organizations become more dependent on software, security-related risks to their organizational missions are also increasing. Traditional security-engineering approaches rely on addressing security risks during the operation and maintenance of software-reliant systems. However, the costs required to control security risks increase significantly when organizations wait until systems are deployed to address those risks. It is more cost effective to address software security risks as early in the lifecycle as possible.

In October 2013, researchers from the CERT® Division at Carnegie Mellon® University's Software Engineering Institute (SEI) started investigating early lifecycle security risk analysis. Our initial research suggests that applying *traditional* security risk-analysis methods earlier in the lifecycle will not solve the problem because those methods cannot handle the inherent complexity of modern cybersecurity attacks. New approaches are needed.

As a result, we developed the Security Engineering Risk Analysis (SERA) Framework, a security risk-analysis approach that advances the existing state-of-the-practice. The SERA Framework incorporates a variety of models that can be analyzed at any point in the lifecycle to (1) identify security threats and vulnerabilities and (2) construct security risk scenarios. Those scenarios are then used to focus an organization's limited resources on controlling the most significant security risks.

This report discusses the contribution of the SERA Framework, given today's increasingly complex threat environment; reviews the framework's basis in existing research and practice; introduces the framework key differentiators; highlights piloting of the framework to elicit better security requirements; and proposes future work to build a SERA method description and additional model types and archetypes to support use of the framework.

1.1 Importance of Software Security

Software assurance is defined as a level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software, throughout the lifecycle [NIA 2010]. Software assurance has been legislatively mandated for the Department of Defense (DoD) in the National Defense Authorization Act for Fiscal Year 2013 [NDAA 2013]. The pursuit of software assurance is a worthy goal that must be translated into practical methods that acquirers, designers, and developers can apply throughout the acquisition-and-development lifecycle.

® CERT and Carnegie Mellon are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Software assurance is becoming increasingly important to organizations across all sectors because of software's increasing influence in business- and mission-critical systems. For example, consider how the size of flight software[1] has increased over the years. Between 1960 and 2000, the degree of functionality provided by software to the pilots of military aircraft has increased from 8% to 80%. At the same time, the size of software in military aircraft has grown from 1,000 lines of code in the F-4A to 1.7 million lines of code in the F-22. This growth trend is expected to continue over time [NASA 2009]. As software exerts more control of complex systems, like military aircraft, the potential risk posed by cybersecurity[2] vulnerabilities will increase in kind.

Cost is another dimension of cybersecurity vulnerabilities that must be taken into account. Many cybersecurity vulnerabilities are considered to be software faults because their root causes can be traced to the software's requirements, architecture, design, or code. Studies have shown that the cost of addressing a software fault increases significantly (up to 200 times) if it is corrected during operations as opposed to design [Mainstay 2010, Microsoft 2014, Soo Hoo 2001]. In addition, rework related to defects consumes more than 50% of the effort associated with a software project. It is thus more cost effective to address software faults early in the lifecycle rather than wait until operations. This principle applies to many operational security vulnerabilities as well.

Operational security vulnerabilities generally have three main causes: (1) design weaknesses,[3] (2) implementation/coding errors, and (3) system configuration errors. Addressing design weaknesses as soon as possible is especially important because these weaknesses are not corrected easily after a system has been deployed. For example, software maintenance organizations normally cannot issue a patch to correct a fundamental security issue related to the software's requirements, architecture, or design. Remediation of design weaknesses normally requires extensive changes to the system, which is costly and often proves to be impractical. As a result, software-reliant systems with design weaknesses often are allowed to operate under a high degree of residual security risk, putting their associated operational missions in jeopardy.

Secure coding and operational security practices help address implementation/coding vulnerabilities and system configuration errors respectively. However, design weaknesses represent 19 of the top 25 weaknesses documented in the Common Weakness Enumeration[4] (CWE) [MITRE 2011]. The importance of design weaknesses in managing cybersecurity risk cannot be overstated.

Our experience indicates that many acquisition and development programs implement compliance-based approaches to address design weaknesses. Engineers typically select security controls based on mandated requirements. However, these compliance-based controls do not necessarily consider the unique characteristics of the operational environment in which a system will be deployed. In addition, attackers are not limited by mandated security controls. They tend to study a

---

1 Flight software is a type of embedded real-time software used in avionics.

2 We use the terms *cybersecurity* and *security* interchangeably in this document.

3 In this report, we define a *design weakness* to be a security-related defect in software's requirements, architecture, or design.

4 The Common Weakness Enumeration (CWE) is an online dictionary of weaknesses that have been found in computer software. The dictionary is maintained by the MITRE Corporation. The purpose of CWE is to facilitate the effective use of tools that identify, find, and resolve bugs, vulnerabilities, and exposures in computer software before the programs are publicly distributed or sold.

system of interest in its operational environment, looking for specific vulnerabilities to exploit. While compliance with mandated security controls is necessary, it might not be sufficient to reduce the residual *operational security risk* of deployed software-reliant systems. Engineers need to look beyond compliance when designing and developing software-reliant systems.

While software and system engineers normally perform extensive analysis to check safety and performance issues, they often do not check for potential problems related to security. By evaluating software-and-system design weaknesses, these engineers can validate that a software-reliant system—as conceived, designed, and built—will perform its functions within an acceptable degree of security risk. Analyzing software security risk early in the acquisition-and-development lifecycle helps build confidence that the system will function as intended during operations even in the event of a cybersecurity attack.

1.2 Focus of the SERA Framework

Prior to developing the SERA Framework, we reviewed relevant research literature to understand the current state of the practice for early lifecycle security risk analysis. Our review indicated that most security risk-analysis methods cannot handle the inherent complexity of modern cybersecurity attacks. Current methods are normally based on a simple, linear view of risk that assumes that a single threat actor exploits a single vulnerability in a single system to cause an adverse consequence. In reality, multiple actors often exploit multiple vulnerabilities in multiple systems as part of a complex chain of events. As a result, we have focused our efforts on developing a risk-analysis approach capable of handling the complexity of today's cybersecurity attacks.

As shown in Figure 1, the SERA Framework incorporates two important technical perspectives: (1) system and software engineering and (2) operational security. The framework requires engineers to consider operational security risks early in the lifecycle. This approach blends multiple technical disciplines and defines a complex analysis activity.

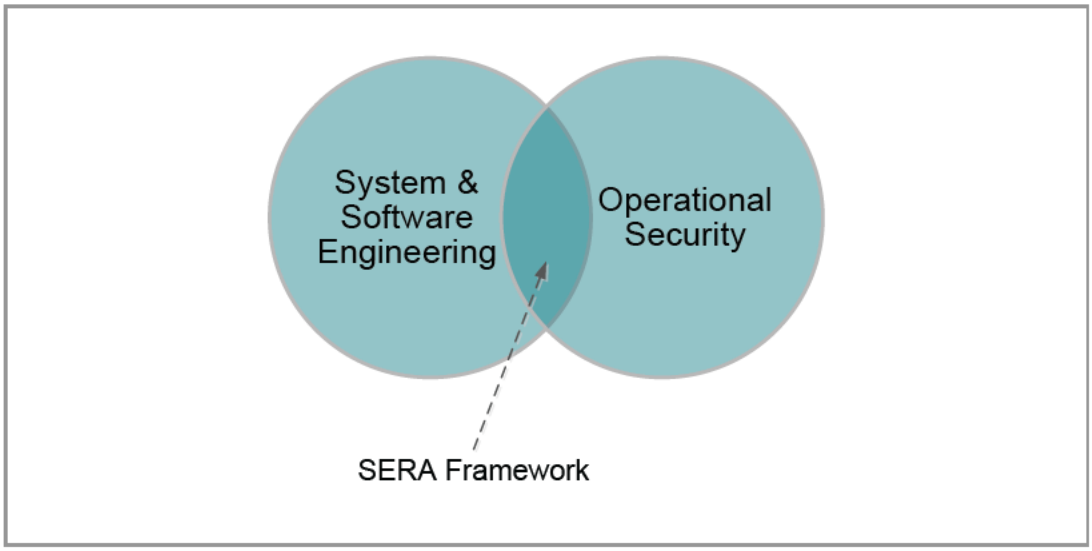

Figure 1: SERA Framework: Technical Perspectives

The SERA Framework is not a quick check-the-box analysis activity. Rather, it defines an engineering practice for analyzing risk in software-reliant systems that are being acquired and developed, with the ultimate goal of building security into those systems. The tasks specified in the framework are designed to be integrated with a program's ongoing system engineering, software engineering, and risk management activities. In fact, our field experience indicates that most programs are already performing many aspects of the framework. By applying the SERA Framework, engineers can assemble and augment existing program information in ways that enable better decisions regarding software security.

1.3 About This Report

This report presents our initial research results, not a final product. The primary audience for this report is anyone interested in learning about new approaches for analyzing security risks during requirements development. In addition, people who are interested in learning about advanced concepts in security risk analysis will also find this document useful. A secondary audience is practitioners, such as systems engineers, software engineers, operational-security risk analysts, and system/software engineering managers. As we mature the SERA Framework, we will develop publications and other products that are oriented toward practitioners. In general, anyone who is interested in the following topics will find this report worthwhile:

- performing early lifecycle security risk analysis
- analyzing security risk in complex environments
- building security into software-reliant systems
- specifying risk-based security requirements

This report provides a conceptual framework for conducting security risk analysis early in the acquisition-and-development lifecycle and presents detailed examples from our early piloting of the framework. This document includes the following sections:

- *Section 1: Introduction*—presents a brief introduction to the SERA Framework and provides some of the motivation for its development
- *Section 2: Problem Space*—defines the six key perspectives of the operational environment in which the SERA Framework is applied
- *Section 3: Security Risk Concepts*—highlights key foundational concepts of cybersecurity risk management
- *Section 4: SERA Approach*—describes the key elements of the security risk environment and highlights the main differentiators of the SERA Framework
- *Section 5: Operational Models*—presents examples of the types of models developed when applying the SERA Framework
- *Section 6: Scenario-Based Risk Analysis*—presents examples of the security risk scenarios produced when applying the SERA Framework
- *Section 7: SERA Framework Overview*—outlines the tasks and steps of the SERA Framework
- *Section 8: Summary and Next Steps*—presents next steps in the development and transition of the SERA Framework
- *Appendix: Example Results for Risk Analysis and Control*—provides examples of analyzed risks and control plans produced when applying the SERA Framework

The main purpose of this report is to present an overview of the SERA Framework. However, before we dive into the details of the framework, we first provide the conceptual basis of our research. The next section of this report begins exploring the fundamental concepts of our research by describing the problem space for our work.

2 Problem Space

Our research-and-development goal for the SERA project is to develop an approach capable of analyzing the complexity of modern cybersecurity risks. Based on this goal, a key question to answer is "What is driving the complexity of security risks?" To answer that question, we begin with the operational environment in which security risk analysis must be performed. Many sources of complexity originate in the network of people, processes, and technologies that form the foundation of an organization and its operational environment. We use the following perspectives to describe the complexity of today's operational environments:

- software
- socio-technical
- cyber-physical
- mission
- system of systems
- compliance

These perspectives are important because they influence how security risk analysis must be performed in practice. As a result, the six perspectives define the problem space for our research-and-development activities. Each perspective is described in the remainder of this section, beginning with the software perspective.

Software Perspective: A *software-reliant system* is a system whose behavior (e.g., functionality, performance, safety, security, interoperability) is dependent on software in some significant way [Bergey 2009]. The software perspective is focused on building security controls into a software-reliant system, not treating security as an add-on feature that will be addressed during software sustainment activities. This perspective requires addressing security concerns from the earliest phases of the system and software lifecycles through the sustainment and evolution of deployed software-reliant systems.

Socio-Technical Perspective: A *socio-technical system* is defined as interrelated technical and social elements that are engaged in goal-oriented behavior. Elements of a socio-technical system include the people who are organized in teams or departments to do their work tasks and the technologies on which people rely when performing work tasks. This perspective stresses the prominent role of people in creating, using, and maintaining technologies. It also highlights the role of people in causing and preventing security attacks.

Cyber-Physical Perspective: A *cyber-physical system* is an engineered system that is built from, and depends upon, the seamless integration of computational algorithms and physical components. Cyber-physical systems merge the physical and virtual worlds, integrating objects, data, and services. Cyber processes monitor and collect data from physical processes, such as the steering of an automobile or the observation of vital signs of a hospital patient. Cyber-physical systems are networked, making their data globally available to other processes. These systems thus make it possible for software to directly interact with events in the physical world. The cyber-physical

perspective emphasizes the notion that cybersecurity attacks can produce consequences in the physical world.

Mission Perspective: A *mission* is a fundamental objective or purpose being pursued by an individual, group, or organization. People, processes, and technologies are then organized in a manner to achieve the mission. This perspective highlights the effect of cybersecurity attacks on the mission that an individual, group, or organization is pursuing. As a result, a security risk analysis must extend beyond the boundary of a technical system and consider the impact on the mission.

System-of-Systems Perspective: A *system of systems* is defined as a set or arrangement of interdependent systems that are related or connected (i.e., networked) to provide a given capability [Levine 2003]. The following characteristics are used to differentiate a system of systems from a very large, complex monolithic system [Maier 1996]:

- *managerial independence*—The management of each system within a system of systems is independent from the management of the other systems.
- *operational independence*—Each system within a system of systems provides useful functionality apart from other systems.
- *evolutionary character*—Each system within a system of systems grows and changes independently of other systems over time.
- *emergent behavior*—Certain behaviors of a system of systems arise from the interactions among the individual systems and are not embodied in any of the individual systems.
- *geographic distribution*—Individual systems within a system of systems are dispersed over large geographic areas.

The system-of-systems perspective describes how a software-reliant system must function as part of a multi-system environment to achieve stakeholders' objectives. This complex, multi-system environment has implications for how security is analyzed and managed. This perspective also illustrates the complex nature of security attacks and how they typically include many systems that are managed by multiple, independent organizational entities.

Compliance Perspective: *Compliance* is defined as the state of being in accordance with established guidelines, specifications, or legislation or the process of becoming so. The compliance perspective describes the range of security guidelines, specifications, and laws to which an organization must adhere. Examples include

- DoD Instruction 8510.01: *Risk Management Framework (RMF) for DoD Information Technology (IT)* [DoD 2014]
- NIST Special Publication 800-37 Revision 1: *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach* [NIST 2010]
- NIST Special Publication 800-30 Revision 1: *Guide for Conducting Risk Assessments* [NIST 2012]
- NIST Special Publication 800-160: *Systems Security Engineering: An Integrated Approach to Building Trustworthy Resilient Systems* [NIST 2014a]
- NIST *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.0* [NIST 2014b]

As a result, a security risk analysis must consider and, when appropriate, incorporate the practices and controls specified in relevant guidelines, specifications, and laws. Each sector may be required to comply with specific and unique set of mandated requirements. However, most mandated cybersecurity requirements share a common set of principles and characteristics.

Traditional security-risk analysis methods generally address one or two of the above perspectives. *An overarching goal of our research is to define a solution that considers all six perspectives*. In the next section, we begin to transition from the problem space to the solution space as we highlight the fundamental concepts of security risk analysis.

3 Security Risk Concepts

The term *risk* is used universally, but different audiences attach different meanings to it [Kloman 1990]. In fact, the details about risk and how it supports decision making depend on the context in which it is applied [Charette 1990]. For example, safety professionals view risk management in terms of reducing the number of accidents and injuries. A hospital administrator views risk management as part of the organization's quality assurance program, while the insurance industry relies on risk management techniques when setting insurance rates. Each industry thus uses a definition that is tailored to its context. No universally accepted definition of risk exists.

Whereas specific definitions of risk might vary, a few characteristics are common to all definitions. For risk to exist in any circumstance, the following three conditions must be satisfied [Charette 1990]:

1. The potential for loss must exist.
2. Uncertainty with respect to the eventual outcome must be present.[5]
3. Some choice or decision is required to deal with the uncertainty and potential for loss.

The three characteristics can be used to forge a basic definition of risk. Most definitions focus on the first two conditions—loss and uncertainty—because they are the two measurable aspects of risk. Thus, the essence of risk, no matter what the domain, can be succinctly captured by the following definition: *Risk is the probability of suffering harm or loss*.[6]

3.1 Security Risk

Security risk is a measure of (1) the likelihood that a threat will exploit a vulnerability to produce an adverse consequence, or loss, and (2) the magnitude of the loss. Figure 2 illustrates the three core components of security risk:

- *Threat*—a cyber-based act, occurrence, or event that exploits one or more vulnerabilities and leads to an adverse consequence or loss
- *Vulnerability*—a weakness in an information system, system security procedures, internal controls, or implementation that a threat could exploit to produce an adverse consequence or loss; a current condition that leads to or enables security risk
- *Consequence*—the loss that results when a threat exploits one or more vulnerabilities; the loss is measured in relation to the status quo (i.e., current state)

From the security perspective, a vulnerability is the passive element of risk. It exposes cyber technologies (e.g., software application, software-reliant system) to threats and the losses that those threats can produce. However, by itself, a vulnerability will not cause an entity to suffer a loss or

[5] Some researchers separate the concepts of certainty (the absence of doubt), risk (where the probabilities of alternative outcomes are known), and uncertainty (where the probabilities of possible outcomes are unknown). However, because uncertainty is a fundamental attribute of risk, this report does not differentiate between decision making under risk and decision making under uncertainty.

[6] This definition is derived from the *Continuous Risk Management Guidebook* [Dorofee 1996].

experience an adverse consequence; rather, the vulnerability makes the entity susceptible to the effects of a threat (adapted from the book titled *Managing Information Security Risks: The OCTAVE[SM] Approach* [Alberts 2006]).

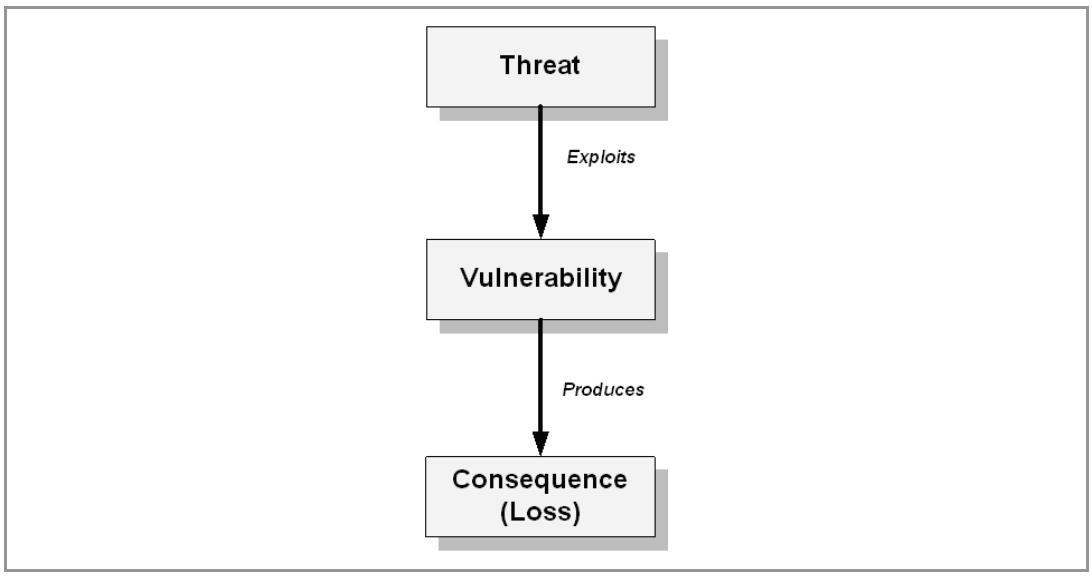

Figure 2: Components of Security Risk

Consider the following example of a security risk. An organization does not encrypt customer data as they are transmitted between systems on the internal network (to ensure quick processing of the data). Malware (i.e., a sniffer), which has been installed in an organization's infrastructure, collects unencrypted customer data (i.e., personally identifiable information) and sends the data to designate staging points across the globe. As a result of this breach in data confidentiality, the organization could suffer significant financial, legal, and reputation consequences.

The components of this security risk are

- *Threat*—Malware collects unencrypted customer data (i.e., personally identifiable information) and sends the data to designate staging points across the globe.
- *Vulnerability*—The organization does not encrypt customer data as they are transmitted between systems on the internal network.
- *Consequence*—The organization could suffer significant financial loss, legal fees, and reputation loss.

In this example, malware exploits a single vulnerability, the unencrypted transmission of data between systems. However, if no threat actor (i.e., malware in the above example) attempts to exploit the vulnerability and carry out the attack, then no adverse consequences will occur. The security vulnerability (e.g., unencrypted data) lies dormant until a threat actor (e.g., malware) attempts to exploit it to produce an adverse consequence or loss.

3.2 Risk Measures

In general, three measures are associated with any risk: (1) probability, (2) impact, and (3) risk exposure.[7] *Probability* is a measure of the likelihood that the risk will occur, and *impact* is a measure of the loss that occurs when a risk is realized. *Risk exposure* provides a measure of the magnitude of a risk based on current values of probability and impact.

3.3 Risk Management

Risk management is a systematic approach for minimizing exposure to potential losses. It provides a disciplined environment for

- continuously assessing what could go wrong (i.e., assessing risks)
- determining which risks to address (i.e., setting mitigation priorities)
- implementing actions to address high-priority risks and bring those risks within tolerance

Figure 3 illustrates the three core risk management activities:

- *Assess risk*—Transform the concerns people have into distinct, tangible security risks that are explicitly documented and analyzed.
- *Plan for controlling risk*—Determine an approach for addressing each security risk; produce a plan for implementing the approach.
- *Control risk*—Deal with each security risk by implementing its defined control plan and tracking the plan to completion.

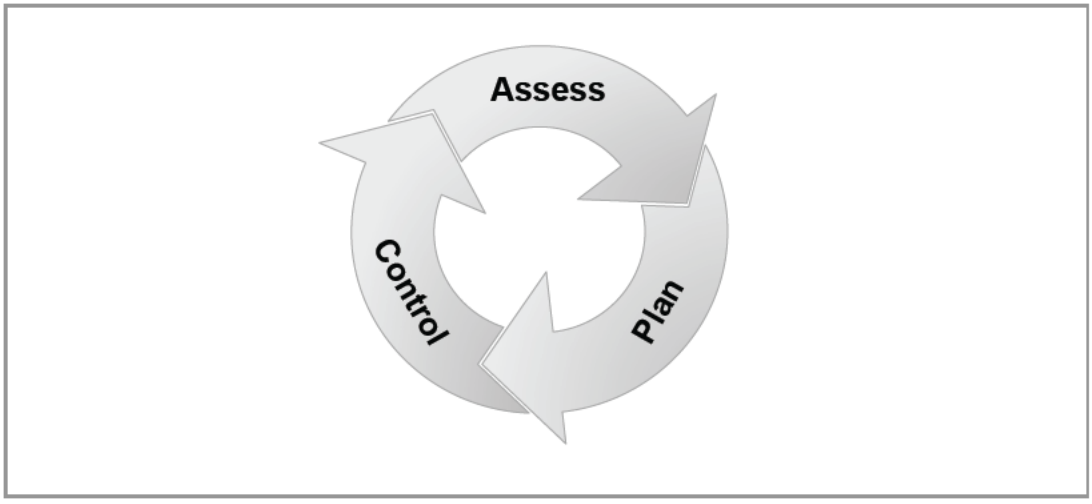

Figure 3: Risk Management Activities

3.4 Controlling Security Risks

The strategy for controlling a risk is based on the measures for the risk (i.e., probability, impact, and risk exposure), which are established during the risk assessment. Decision-making criteria (e.g., for prioritizing risks or deciding when to escalate risks within an organization) may also be

[7] A fourth measure, *time frame*, is sometimes used to measure the length of time before a risk is realized or the length of time in which action can be taken to prevent a risk.

used to help determine the appropriate strategy for controlling a risk. Common control approaches include

- *Accept*—If a risk occurs, its consequences will be tolerated; no proactive action to address the risk will be taken. When a risk is accepted, the rationale for doing so is documented.
- *Transfer*—A risk is shifted to another party (e.g., through insurance or outsourcing).
- *Avoid*—Activities are restructured to eliminate the possibility of a risk occurring.
- *Mitigate*—Actions are implemented in an attempt to reduce or contain a risk.

For any security risk that is not accepted, the security analyst should develop and document a control plan for that risk. A control plan defines a set of actions for implementing the selected control approach. For risks that are being mitigated, their plans can include actions from the following categories:

- *Recognize and respond* – Monitor the threat and take action when it is detected.
- *Resist* – Implement protection measures to reduce vulnerability to the threat and minimize any consequences that might occur.
- *Recover* – Recover from the risk if the consequences or losses are realized.

Thus far in this section, we provide a simplified view of security risk, where a single threat actor exploits a single vulnerability in a single system to cause an adverse consequence. Most traditional security risk analysis methods are based on this simplified view of risk. However, in reality, multiple actors exploit multiple vulnerabilities in multiple systems as part of a complex chain of events. In the next section, we look at the inherent complexity of security risk.

3.5 Complexity of Security Risk

Consider the following example of a complex risk scenario. In this scenario, an individual (i.e., the perpetrator) intends to steal personally identifiable information about an organization's customer base. The individual's goal is to steal the identities of customers for financial gain. To carry out this risk scenario successfully, the individual performs the following actions:

- The individual performs reconnaissance on the organization's systems and networks.
- The individual also performs reconnaissance on partners and collaborators that work with the organization and have trusted access to the organization's systems and networks.
- Reconnaissance indicates that the organization has strong perimeter security controls in place. As a result, the individual targets a third-party collaborator that (1) has legitimate, trusted access to the organization's internal network and (2) has relatively weak perimeter security controls in place.
- The individual gains access to the third-party collaborator's internal network by exploiting several common vulnerabilities.
- The individual uses the collaborator's trusted access to the organization's internal network to bypass the organization's perimeter security controls and gain access to its network.
- Additional reconnaissance indicates that the organization does not encrypt customer data as they are transmitted between an order entry system and an inventory system (to ensure quick processing of the data). In addition, the organization does not employ rigorous monitoring in its systems and networks. The organization's strategy is to focus primarily on its perimeter

security. The individual decides to exploit these vulnerabilities and installs malware (i.e., a sniffer) that is designed to

  - steal unencrypted customer data as it is being transmitted between systems on the internal network
  - send the stolen data to staging points at multiple external locations

- Once installed, the malware collects unencrypted data and sends the data to the staging points. This data exchange is timed to occur during peak business hours to mask the attack.

As a result of this scenario, the organization could suffer significant financial, legal, and reputation consequences. The crux of this scenario is identical to the risk that we highlighted in Section 3.1; however, the risk scenario presented in this section is considerably more complex. This risk scenario better represents the inherent complexity of modern security attacks, where multiple actors[8] exploit multiple vulnerabilities[9] in multiple systems[10] as part of a complex chain of events. Traditional methods are often unable to analyze complex security attacks effectively. Our research is intended to address this deficiency in traditional security risk-analysis methods.

---

[8] In the scenario, both the individual that initiates the attack and the malicious code are considered to be threat actors.

[9] Vulnerabilities in the scenario include lack of monitoring to detect the actor's reconnaissance activities; allowing trusted access to the organization's internal network by an third-party collaborator that employs poor security practices; the organization's lack of rigorous monitoring of its systems and networks; and lack of data encryption between the order entry and inventory systems.

[10] Systems involved in the attack include system owned by the third-party collaborator, order entry system, inventory system, perimeter security systems/devices, and various networking systems/devices.

4 SERA Approach

The example from Section 3.5 illustrates the notion that modern security risks are highly complex. In our current research, we are developing the SERA Framework to analyze complex security risks. We begin this section by describing the key elements of the security risk environment in which our analysis approach is applied. We then close out this section by highlighting two key differentiators of the SERA Framework.

4.1 Security Risk Environment

In Section 2, we describe the problem space for our research-and-development activities. Within that context, we take a closer look at the security risk environment and its characteristics. Figure 4 depicts our conceptual view of the security risk environment.

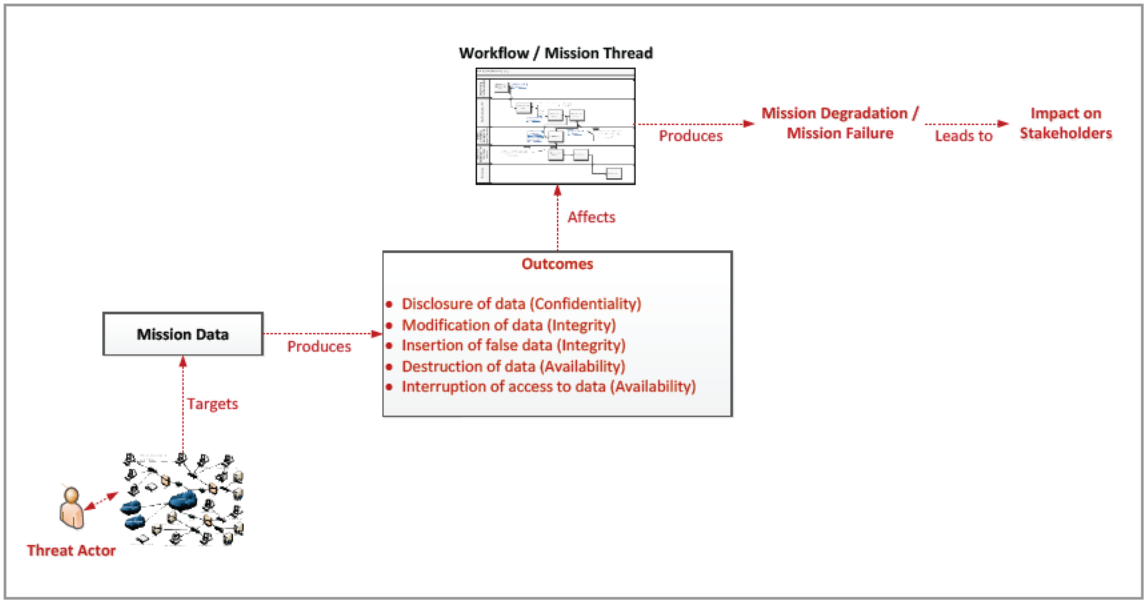

Figure 4: Security Risk Environment

The focal point of the environment is the threat actor. A common goal of many threat actors is to inflict harm or loss on a mission's stakeholders. To accomplish that goal, a threat actor first targets data that are used to support a workflow or mission thread.[11] To access targeted mission data, a threat actor must navigate through the complex network of people, processes, and technologies, looking for weaknesses in organizational security practices (*socio-technical perspective*) and vulnerabilities in software-reliant systems to exploit (*software perspective*). Getting to the mission

[11] A *workflow* is a collection of interrelated work tasks that achieves a specific result [Sharp 2001]. A workflow includes all tasks, procedures, organizations, people, technologies, tools, data, inputs, and outputs required to achieve the desired objectives. The business literature uses several terms synonymously with workflow, including work process, business process, and process. *Mission thread* is essentially the term the military uses in place of *workflow*. A *mission thread* is a sequence of end-to-end activities and events that takes place to accomplish the execution of a military operation. In this document, we use the terms *workflow* and *mission thread* synonymously.

data can be difficult. A threat actor may need to jump from one targeted computer to another when attempting to realize the goal of the attack. In many cases, an actor may target computers that are owned and maintained by trusted partners and third-party collaborators when conducting a cyber-attack (*system-of-systems perspective*).

The threat actor is ultimately looking to violate the security attributes of mission data, with the hope of causing a range of indirect, negative consequences for mission stakeholders. Data have three basic security attributes: confidentiality, integrity, and availability.[12] For a given risk, a threat actor generally is trying to produce one or more of the following outcomes:

- disclosure of data (violation of the confidentiality attribute)
- modification of data (violation of the integrity attribute)
- insertion of false data (violation of the integrity attribute)
- destruction of data (violation of the availability attribute)
- interruption of access to data (violation of the availability attribute)

Each outcome maps to a security attribute of the data. The example from Section 3.5 illustrates a scenario in which a threat actor steals unencrypted customer data as they are being transmitted between systems on an internal network. The customer data should only be viewed by people within the company who have been authorized to view it. As a result, the confidentiality attribute of the customer data is violated because the threat actor is not authorized to view that data.

In that example, the threat actor is targeting an organization's order entry and inventory workflow. A key part of the organization's mission is to protect personally identifiable information from being viewed (and stolen) by unauthorized parties (*mission perspective*). Protecting customer information is part of an effective organizational security program; however, in many cases, the protection of personally identifiable information is also mandated by laws and regulations (*compliance perspective*). As indicated in Figure 2, the violation of a security attribute has an impact on the workflow/mission thread and its ability to achieve its mission successfully.

The final basic element of the security risk environment is the impact on mission stakeholders.[13] When a threat actor produces mission degradation or mission failure, the consequence can have a negative impact on various stakeholder groups. The example from Section 3.5 could lead to many adverse consequences. The identities of customers could be stolen, leading to considerable personal financial losses. Those consumers could sue the company for not protecting their personal data properly, resulting in legal fees and resulting financial penalties for the company. Also, the company could suffer a loss in reputation, which could adversely affect its profits. This example illustrates one type of attack that could be directed at an organization's order entry and inventory management processes. Other types of cyber-attacks could produce very different consequences,

---

[12] *Confidentiality* is defined as keeping proprietary, sensitive, or personal information private and inaccessible to anyone who is not authorized to see it. *Integrity* is defined as the authenticity, accuracy, and completeness of data. *Availability* is defined as the extent to which, or frequency with which, data mist be present or ready for use. These definitions are adapted from the book titled *Managing Information Security Risks: The OCTAVE[SM] Approach* [Alberts 2002].

[13] A *stakeholder* is defined in this document as a person or group with an interest in a workflow/mission thread and the products it produces or the services it provides.

such as sending the wrong merchandise to customers (*cyber-physical perspective*). Organizations must be prepared to guard against a range of cyber-attacks.

The conceptual view of the security risk environment highlights the complex nature of security risks. The ultimate focus of our SERA research-and-development project is to develop a systematic means for sorting through this complexity and enabling effective decision making. This focus has led to a unique approach for analyzing security risk in complex environments.

4.2 SERA Differentiators

The SERA Framework incorporates two key design features that differentiate it from other security risk assessments. The first is the use of operational models. Participants applying traditional security-risk assessments typically rely on their tacit understanding of the operational context in which a software-reliant system must operate. Our experience indicates that tacit assumptions are often incorrect or incomplete, which adversely affects the results of a security risk analysis. We propose using operational models to describe a system's operational context explicitly. This topic is the focus of Section 5 of this report.

The second feature is the semantics that we have defined to document security risks. Most traditional assessments rely on linear, simplistic structures for recording risks. These methods are based on the premise that a single threat actor exploits a single vulnerability in a single system to cause an adverse consequence. For example, many of these methods commonly employ an *if-then statement* to capture a risk. The *if* part of the statement coveys how the threat exploits a vulnerability, while the *then* portion expresses the resulting consequence. However, basic formats, such as the if-then statement, are too simplistic to capture that complexity of modern cybersecurity attacks. Here, we propose using scenarios to document the inherent complexities and nuances of security risk. We describe the structure of security risk scenarios in Section 6 of this report.

5 Operational Models

Many risk-identification methods are based on brainstorming techniques. Participants describe risks based on their tacit understanding of the operational environment. For security risk-identification methods, people tend to identify threats with which they have some familiarity. They also tend to describe consequences based on their personal knowledge of organizational workflows and associated stakeholders. In lieu of brainstorming, we propose that people conduct a detailed analysis that employs a multi-model approach for establishing operational content.

5.1 Multiple Models

Most traditional risk-identification methods do not explicitly describe the operational environment. As a result, each participant in the brainstorming session relies on his or her mental model of the environment. Each person is relying on his or her assumptions that are likely to be incorrect, incomplete, or in conflict with the assumptions of other participants. These participants do not have a common view of the operational environment. This is especially problematic when security risks are being identified early in the lifecycle. The environment might not be well described or documented, which makes people's perspectives vary widely.

To counteract this lack of a common perspective, we propose developing models that describe the operational environment in which the system will be deployed. Table 1 provides a description of the operational views that we have been using in our pilot activities of the SERA Framework. Each view is characterized using one or more models.

Table 1: Operational View

View	Description
Workflow/Mission Thread	The sequence of end-to-end activities and events that take place to achieve a specific result
Stakeholder	The set of people with an interest or concern in (1) the workflow/mission thread and (2) the outcomes (e.g., products, services) produced by it
Data	The data items that are required when executing the workflow/mission and their associated security attributes (confidentiality, integrity, availability)
Network	The projected network topology for the system of interest
Physical	The projected physical layout of the facilities in which components of the system of interest are located
Use Case	A description of a set of steps that define the interactions between a role/actor and a system to achieve a goal (The actor can be a human or an external system.)

Developing and documenting operational models enables analysts to address aspects of complexity that are inherent in the security risk environment. (See Section 4.1 for a description of the security risk environment.) Models representing the views from Table 1 can be analyzed to establish the following key aspects of a threat:

- *critical data*—important information highlighted in workflow/mission thread, use case, and network diagrams. By examining these models, analysts can identify which data elements are most critical to the workflow/mission thread and its associated mission.
- *access path*—how a threat actor can gain access to data and violate its security attributes (i.e., create breaches of data confidentiality, integrity, and availability). The network and physical models provide insights into potential cyber and physical access paths for an attack.
- *threat outcome*—the direct consequence caused by the threat. A direct consequence describes which security attributes of critical data have been breached. Examples of outcomes include data disclosure, data modification, insertion of false data, destruction of data, and interruption of access to data. The data model is used to identify the immediate consequence of a threat.

A threat ends with a description of its direct consequence or outcome. However, a security risk analysis must also take into account any indirect consequences triggered by the occurrence of a threat. For example, if false data are inserted into a workflow or mission thread, then the following questions related to indirect consequences must be answered:

- How is the workflow/mission thread affected?
- How are the mission's objectives affected?
- How are mission's stakeholders affected?

The indirect consequences are used to (1) measure the impact of a security risk and (2) establish a risk's priority for decision makers. Analysts determine indirect consequences using models that represent the workflow/mission thread and stakeholder views. In the remainder of this section, we provide examples of three operational models that we developed for our initial pilot of the SERA Framework: (1) top-level workflow model, (2) top-level network model, and (3) data model.

5.2 Example: Operational Models

For our pilot application of the SERA Framework, we analyzed security risk in the Wireless Emergency Alerts (WEA) service, which is a collaborative partnership that includes the cellular industry, Federal Communications Commission, Federal Emergency Management Agency (FEMA), and U.S. Department of Homeland Security (DHS) Science and Technology Directorate (S&T) [WEA 2014]. The WEA service enables local, tribal, state, territorial, and federal public safety officials to send geographically targeted text alerts to the public to warn it about emergency situations. The best place to start when describing the WEA service is with its workflow.

5.2.1 Top-Level Workflow Model

An *emergency alert* is a message sent by an authorized organization that provides details of an occurring or pending emergency situation to one or many designated groups of people. Emergency alerts are initiated by many diverse organizations. For example, law enforcement organizations issue amber alerts, and the National Weather Service (NWS) issues weather alerts. Both amber alerts and weather alerts are examples of emergency alerts. A *wireless alert* is an emergency alert

that is sent to mobile devices, such as cell phones and pagers. Figure 5 shows the top-level workflow model that we developed for the WEA service.[14]

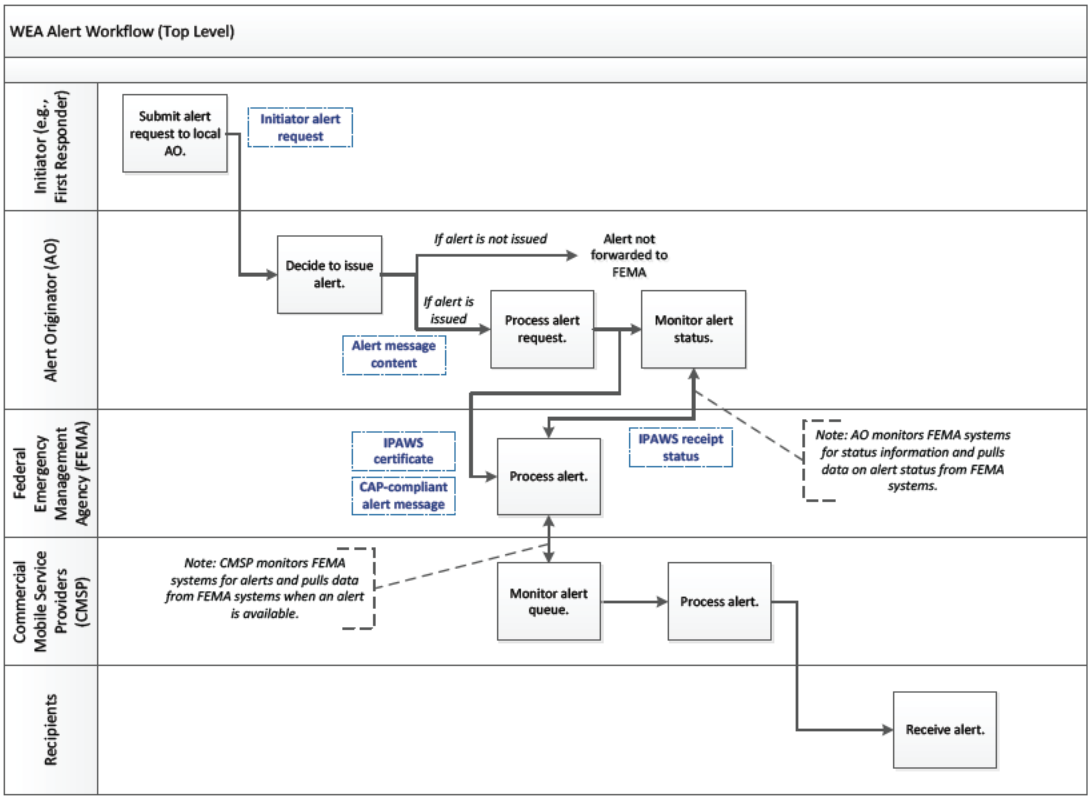

Figure 5: Top-Level Workflow Model

The figure shows the sequence of activities required to issue a wireless alert. We used a swimlane diagram[15] to document the workflow. The activities in a swimlane diagram are grouped visually by placing them in *lanes*. Parallel lines divide the diagram into multiple lanes, with one lane for each workflow actor (i.e., person, group, or sub-process). Each lane is labelled to show who is responsible for performing the activities assigned to that lane. In Figure 5, the gray boxes with solid borders represent the activities that are performed by each workflow actor. Lines between the activities establish the relationships among and sequencing of the activities. Finally, the blue, dashed boxes are data items that flow between the activities.

The workflow begins with a request from an initiator (e.g., law enforcement, NWS) to submit an alert (*initiator alert request*). A team from the Alert Originator (AO) organization receives the ini-

[14] We also developed a second-level workflow that contained more details about the WEA service as part of our analysis. Because of the limited scope of this document, we are only showing the top-level workflow in this report.

[15] A swimlane diagram provides a visual representation of a workflow or mission thread. It defines the sequence of end-to-end activities that take place to achieve a specific result as well as who performs each activity. Swimlane diagrams are especially useful for describing workflows or mission threads that cross organizational boundaries, which a characteristic of system-of-systems environments. Because we are focusing on system-of-systems environments in our research, we have found swimlane diagrams to be a useful workflow modeling technique.

tiator alert request and decides (1) whether or not to issue the alert and (2) the distribution channels for the alert (e.g., television, radio, roadside signs, wireless technologies, others). The workflow in Figure 5 assumes a wireless alert will be issued.

An operator from the AO enters the *alert message content* into an Alert Originating System (AOS), which then processes the content. The AOS converts the alert message to a CAP-compliant format,[16] which is the data format required by FEMA systems. The *CAP-compliant alert message* and *IPAWS certificate* are then sent from the AOS to a FEMA system named IPAWS-OPEN Gateway.[17] The IPAWS certificate is used to establish that the CAP-compliant alert message is being sent from a legitimate AO. The certificate is encrypted during transmission to IPAWS-OPEN Gateway while the CAP-compliant alert message is not encrypted. (The CAP-compliant alert message is considered to be public information and thus does not need to be encrypted. Table 2 highlights the security attributes of the CAP-compliant alert message and IPAWS certificate.)

The IPAWS-OPEN Gateway decrypts the IPAWS certificate and validates the identity of the sender. If the certificate is determined to be valid, the IPAWS-OPEN Gateway

- logs receipt of the alert message (*IPAWS receipt status*) in the IPAWS-OPEN log
- immediately forwards the alert message to other FEMA systems for processing

The AO can monitor the IPAWS-OPEN log to make sure that the IPAWS-OPEN gateway received the CAP-compliant alert message.

FEMA systems then process the wireless alert and forward it to the commercial mobile service providers (CMSPs). AT&T, Verizon, and other wireless carriers are examples of CMSPs. The CMSP systems process and format the alert message and then distribute it to recipients' smart phones. Finally, recipients receive and read the wireless alert on their smart phones.

5.2.2 Top-Level Network Model

Figure 5 features a top-level workflow that describes the core activities needed to distribute an emergency alert using the WEA service. The workflow provides the anchor for the subsequent security risk analysis. After we develop a workflow model, we then determine which technologies support that workflow.

The systems that support the WEA workflow are shown in Figure 6. In essence, the collection of systems in Figure 6 depicts the WEA system of systems. These systems support the end-to end WEA workflow and are the starting point for a deep dive into an analysis of WEA support technologies.[18]

The following are the highlights of the WEA system of systems depicted in Figure 6:

---

[16] Common Alerting Protocol (CAP)

[17] Integrated Public Alert and Warning System Open Platform for Emergency Networks (IPAWS-OPEN)

[18] We developed additional, more detailed network diagrams as part of our analysis. Because of the limited scope of this document, we are only showing the top-level network model in this report.

- *Initiator systems*—Communication of alert information between the initiator and AO can use the following technologies: telecommunications (for verbally communicating requests) and unencrypted email from the initiator's desktop computers.
- *AO systems*—The AO uses three systems: telecommunications, AO desktop computers, and the AOS. The AO relies on the following technologies to receive requests to issue an alert: telecommunications (for verbally receiving requests) and unencrypted email sent from an initiator's desktop computer to AO desktop computers.[19] After AO management decides to issue a wireless alert, an AO operator enters the alert into the AOS, which then forwards the CAP-complaint alert message to the IPAWS-OPEN Gateway (i.e., a FEMA system).
- *FEMA systems*—The IPAWS-OPEN Gateway receives the alert message, validates the sender using the certificate, and forwards the alert to the WEA Aggregator for processing. The WEA Aggregator processes the wireless alert and transmits it to the Federal Alert Gateway, which then sends the alert message to CMSP Gateway.
- *CMSP systems*—The CMSP Gateway receives the alert message and then forwards it to CMSP Infrastructure (e.g., cell towers). The alert message is transmitted by the CMSP Infrastructure to capable wireless devices in the designated area(s).
- *Recipient systems*—People in the designated area(s) that have devices capable of receiving wireless alerts receive the message on their wireless devices.

[19] Data from AO desktop computers cannot be sent over the network to the AOS. Operators must use removable media, such as USB drives, to exchange data between these two systems.

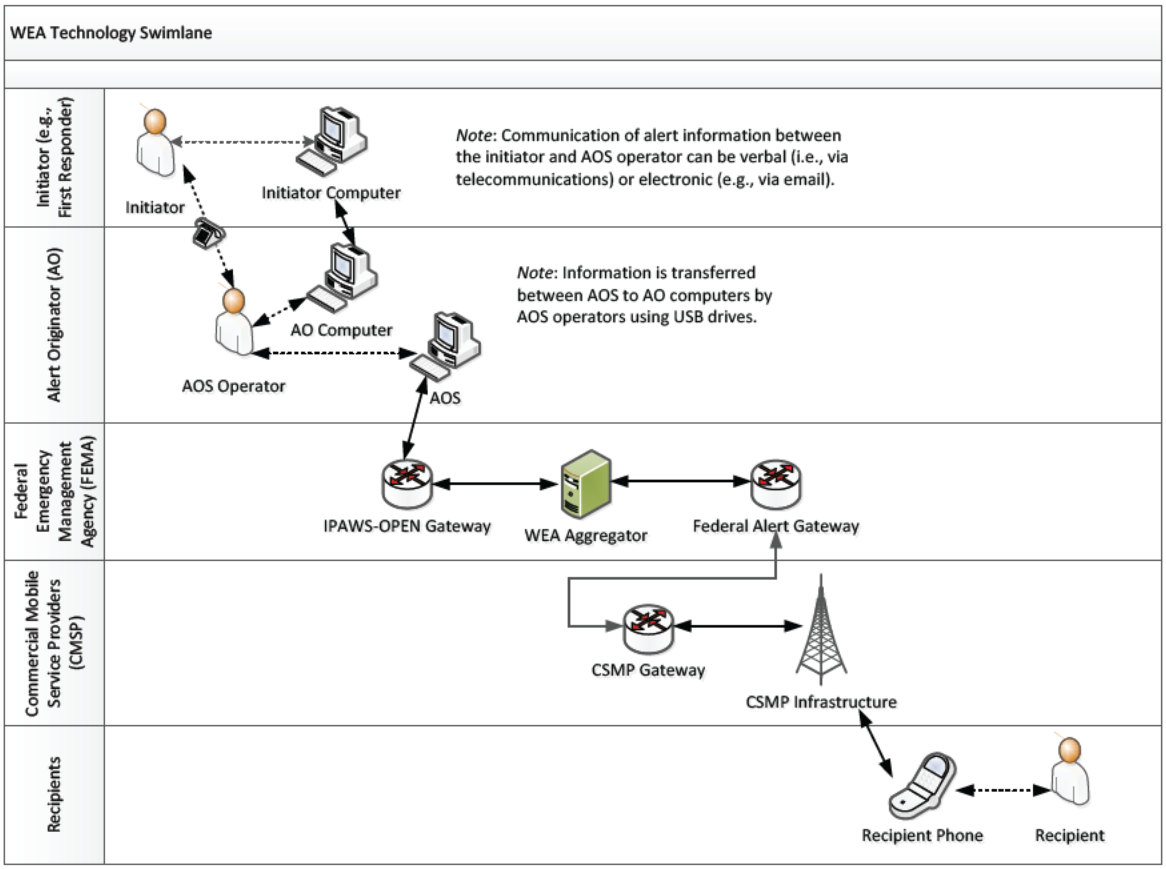

Figure 6: Top-Level Network Model

For our pilot application of the SERA Framework, the pilot's stakeholders identified the AOS as the system of interest for the analysis. Here, we define the *system of interest* as the software application or system that is the focus of the security risk analysis.[20] The operational environment for the system of interest is then characterized to establish a baseline of operational performance. Security risks are analyzed in relation to this baseline. We developed several models related to the AOS, including

- detailed workflows
- detailed network diagrams
- projected facility layout of AOS computers
- data security attributes for AOS data
- AOS use cases

We did not include all of the above models in this report because they are beyond the scope of the introductory nature of this document. We plan to provide a more detailed treatment of the AOS models in subsequent publications. We conclude this section with a look at a model that describes the security attributes of critical AOS data.

[20] Figure 6 shows the relationship between the AOS and the WEA system of systems.

5.2.3 Data Model

From the system-of-systems perspective, we identified three critical data items (i.e., critical assets) for the AOS:[21]

1. *initiator alert request*—a request to submit an alert sent from an initiator computer to the AOS
2. *CAP-compliant alert message*—the alert message in CAP-compliant format sent from the AOS to the IPAWS-OPEN Gateway
3. *IPAWS certificate*—sent from the AOS to the IPAWS-OPEN Gateway along with the *CAP*-compliant alert message and used to establish that the alert message was sent from a legitimate AO

Each of the three data items appears on the top-level workflow and is stored, transmitted, or processed by the system of interest (i.e., the AOS). Table 2 presents the security attributes (i.e., confidentiality, integrity, and availability) for the three critical data items. The security attributes are essential input when identifying threats to the AOS.

Table 2: WEA Data Model

Data Element	Form	Confidentiality	Integrity	Availability
Initiator alert request	Verbal or Electronic	There are no restrictions on who can view this data element. (public data)	The data element must be correct and complete. (high data integrity)	This data element must be available when needed. (high availability)
CAP-compliant alert message	Electronic	There are no restrictions on who can view this data element. (public data)	The data element must be correct and complete. (high data integrity)	This data element must be available when needed. (high availability)
IPAWS certificate	Electronic	Only authorized people can view this data element. (sensitive but unclassified)	The data element must be correct and complete. (high data integrity)	This data element must be available when needed. (high availability)

Our intent for this section is to provide an overview of the types of operational models that help enable effective risk identification. Our field experience indicates most risk-identification methods rely on participants' tacit understanding of the operational environment, which is often incorrect, incomplete, or in conflict. Our goal is to use operational modeling to construct a common view of the projected operational environment. From this common, or baseline, view of the operational environment, analysts can then develop a set of relevant security risk scenarios. In the next section, we present our prototype structure for describing security risk scenarios.

[21] The three data items were identified by analyzing the top-level workflow from Figure 5 and the top-level network diagram from Figure 6. Additional data items were identified by analyzing detailed AOS models that we developed but have not included in this report. Because of the limited scope of this document, we are only showing the data items in Table 2.

6 Scenario-Based Risk Analysis

The second key differentiator of the SERA approach is the use of scenarios to describe security risks. Our early piloting of the SERA Framework indicates that scenarios capture the complexities and nuances of a security risk. We define a *security risk scenario* as a narrative description of a complex security risk. A security risk scenario tells a story of how one or more threat actors can cause adverse consequences for stakeholders by exploiting vulnerabilities in one or more software-reliant systems.

We develop a scenario after performing a detailed analysis of a threat, its enablers, and the consequences it can produce. The development of a security risk scenario is thus the culmination of an extensive analysis activity. The SERA Framework requires that the following data are recorded for each security risk:

- security risk scenario
- risk statement
- threat components
- threat sequence
- workflow consequences
- stakeholder consequences
- enablers

In this section, we present examples of the above data in the context of a security risk to the WEA service.

6.1 Security Risk Scenario

The following scenario describes a spoofing attack[22] that targets an AOS:

> An outside actor with malicious intent plans to obtain a valid IPAWS certificate through social engineering[23] and then use it to send an illegitimate CAP-compliant alert message to the IPAWS-OPEN Gateway. In carrying out this attack, the actor plans to spoof an AOS. First, the threat actor performs reconnaissance to gather the information needed to
>
> - conduct social engineering to get a valid IPAWS certificate and the associated encryption key from an AO
> - construct an illegitimate CAP-compliant alert message that will be accepted by the IPAWS-OPEN Gateway
>
> The threat actor identifies several social engineering targets based on the results of reconnaissance. The actor performs social engineering on people at the AO that have access to the

[22] A *spoofing attack* is a circumstance in which a person or program successfully impersonates another person or program by falsifying data. The person or program perpetrating the attack is free to take action that will be attributed to the victim of the spoofing attack.

[23] *Social engineering* refers to the intentional manipulation of people to get them to perform certain actions or divulge confidential information.

AO's IPAWS certificate and encryption key. After several attempts, the actor obtains an AO's valid IPAWS certificate and encryption key from an unsuspecting employee at the AO. The actor gains access to data specifications for constructing CAP-compliant alert messages from public documents.

Now that the actor has a valid IPAWS certificate and knows how to format data for the IPAWS-OPEN Gateway, he or she can execute the attack. The actor's goal is to incite panic in a crowd that a bomb is about to explode (e.g., an alert message of a bomb about to explode in Times Square on New Year's Eve or at a major sporting event). The actor will send an illegitimate alert message to the wireless devices of people in the crowd. To maximize the impact of the attack, the actor must send the false alert when a large crowd has gathered for an event.

To send an illegitimate alert message, the threat actor

- constructs an illegitimate CAP-compliant alert message
- links it to the IPAWS certificate
- encrypts the IPAWS certificate
- sends the illegitimate, unencrypted CAP-compliant alert message and encrypted IPAWS certificate to the IPAWS-OPEN Gateway

The IPAWS-OPEN Gateway decrypts the IPAWS certificate and validates the identity of the sender. The certificate is determined to belong to a valid AO, and the IPAWS-OPEN Gateway

- logs receipt of the alert message in the IPAWS-OPEN log
- immediately forwards the alert message to other FEMA systems for processing

Staff members from the AO are not monitoring the IPAWS-OPEN log because they have not sent an alert message. The FEMA systems process the illegitimate alert message and place it in the CMSP alert queue. CMSP systems are monitoring the CMSP alert queue and grab the alert message. Next, CMSP systems distribute the alert to recipients' wireless devices. Recipients receive and read the illegitimate alert on their wireless devices. At this point, staff members from the AO are still unaware that a threat actor has spoofed their AOS and delivered an illegitimate wireless alert to a group of people.

This scenario could have considerable impact on stakeholders, depending on the severity of the event with which the attack is linked. Health and safety damages could be significant, leading to potentially large legal liabilities. Such an attack could damage the reputation of the WEA service beyond repair.

The above example illustrates how a threat actor can execute an AOS spoofing attack that targets the WEA service. The purpose of the scenario is to convey the most important aspects of a security risk; it does not include all of the details needed to analyze the risk. As a result, we record additional data for each scenario. In the remainder of this section, we present examples of these additional data for the AOS spoofing risk.

6.2 Risk Statement

As we note in Section 4.2, many traditional risk assessments use if-then statements to represent a security risk. Those assessments rely on the if-then structure to convey all relevant information about a security risk. In contrast, the SERA Framework uses a risk statement as a shorthand description of a security risk scenario. For example, the following risk statement describes the AOS spoofing risk from Section 6.1:

> **IF** an outside actor with malicious intent obtains a valid certificate through social engineering **AND** uses it to send an illegitimate CAP-compliant message by spoofing an AOS, **THEN** health, safety, legal, financial, and reputation consequences could result.

The SERA Framework requires analysts to use the security risk scenario and supporting data structures (i.e., not the summary if-then statement) when analyzing security risks and making decisions about how to control them. Risk statements are used to facilitate the tracking of multiple security risk scenarios during analysis and control. [24] (Refer to Table 16 in the appendix for an example of how risk statements are embedded in a risk tracking spreadsheet.)

6.3 Threat Components

When constructing risk scenarios, we develop and document considerable information about the underlying threat. We refer to this information as threat components because they examine different facets of a threat. Threat components provide additional details that are not part of the risk statement and might not be conveyed in the security risk scenario. Threat components include the following items:

- *threat*—a statement that describes the cyber-based act, occurrence, or event that exploits one or more vulnerabilities and leads to an adverse consequence or loss (The threat statement provides the content for the *if* portion of the risk statement.)
- *actor*—who or what is attempting to violate the security attributes of critical data
- *motive*—the intentions of a threat actor, which can be deliberate/malicious or accidental
- *goal*—the end toward which the threat actor's effort is directed (The goal succinctly describes the key indirect consequence [i.e., impact on stakeholders] that the actor is trying to produce.)
- *outcome*—the direct consequence of the threat (i.e., disclosure of data, modification of data, insertion of false data, destruction of data, interruption of access to data)
- *means*—the resources the actor uses when executing the threat
- *threat complexity*—the degree of difficulty associated with executing the threat
- *additional context*—any additional, relevant contextual information related to the threat

Table 3 highlights the threat components for the AOS spoofing risk.

---

[24] During our pilots of the SERA Framework, we analyzed multiple security risk scenarios. We assigned an identifier to each risk statement and put all risk statements (and associated identifiers) into a spreadsheet. After evaluating each scenario's probability, impact, and risk exposure, we added those values to the spreadsheet as well. We then prioritized the security risk scenarios based on their risk measures (probability, impact, and risk exposure). The risk statement provided a succinct way of differentiating the security risk scenarios in the spreadsheet. Table 16 in the appendix of this document provides an example of a risk tracking spreadsheet.

Table 3: Threat Components

Component	Description
Threat	An outside actor with malicious intent obtains a valid certificate through social engineering and uses it to send an illegitimate CAP-compliant message by spoofing an AOS.
Actor	An actor is a person with an outsider's knowledge of the organization.
Motive	The threat is a deliberate/malicious act.
Goal	To incite panic in a crowd that a bomb is about to explode (e.g., an alert message of a bomb in Times Square on New Year's Eve).
Outcome	False data are sent to the IPAWS-OPEN Gateway for processing. (integrity issue)
Means	The actor only needs a networked computer and access to public documents that describe the WEA service.
Threat Complexity	The attack is complex and requires significant preparation to execute.
Additional Context	The actor needs to time the attack to coincide with an event where a large crowd will gather.

6.4 Threat Sequence

The threat sequence describes the series of actions taken by the actor(s) when executing the threat. Table 4 illustrates the ten steps needed to produce the threat underlying the AOS spoofing risk. We use the threat sequence when developing the narratives of a security risk scenario. (See Section 6.1 for the AOS spoofing scenario.)

Table 4: Threat Sequence

Step	
1.	The actor performs reconnaissance to determine who to target for social engineering.
2.	The actor selects an appropriate target for social engineering (e.g., an employee at the AO, vendor, or FEMA that has legitimate access to the AO certificate and the associated encryption key).
3.	The actor conducts a social engineering attack on the employee to obtain the AO certificate and encryption key.
4.	The employee provides an electronic copy of the certificate and encryption key to the actor.
5.	The actor finds information about constructing CAP-compliant messages from public documents.
6.	The actor creates an illegitimate CAP-compliant message (i.e., illegitimate wireless alert) intended to incite panic in a crowd that a bomb is about to explode (e.g., an alert message of a bomb that is about to explode in Times Square on New Year's Eve or at a major sporting event).
7.	The actor sends the illegitimate CAP-compliant message (unencrypted) and certificate (encrypted) to the IPAWS-OPEN Gateway.
8.	The IPAWS-OPEN Gateway decrypts the AO certificate and validates the identity of the sender.
9.	The IPAWS-OPEN Gateway logs receipt of the alert message.
10.	The IPAWS-OPEN Gateway processes the alert by sending it to the WEA aggregator.

6.5 Workflow/Mission Thread Consequences

A threat produces a direct consequence, which is called the outcome of the threat. A threat's outcome indicates how the security attributes of critical data are violated; it does not indicate the potential impact on the objectives of the workflow or mission thread. To fully analyze a threat's impact, analysts must look beyond its direct consequence and examine how the threat might affect the projected operational environment. This process begins by examining how the outcome (i.e., direct consequence) might affect the objectives of the workflow or mission thread (i.e., indirect consequence of the threat's occurrence). Table 5 provides the details of the workflow consequences for the AOS spoofing risk.

Table 5: Workflow Consequences

	Workflow Consequence
1.	The IPAWS-OPEN Gateway decrypts the AO certificate and validates the identity of the sender.
2.	The IPAWS-OPEN Gateway logs receipt of the alert message in the AOS log[25] and forwards the alert message to other FEMA systems for processing.
3.	FEMA systems place the alert message in the CMSP alert queue.
4.	CMSP systems monitor the CMSP alert queue and grab the alert message.
5.	CMSP systems distribute the alert to recipients' smart phones.
6.	Recipients receive and read the illegitimate alert on their smart phones.

To execute this threat, the actor spoofs an AOS and sends an illegitimate alert message and encrypted IPAWS certificate to the IPAWS-OPEN Gateway. We analyzed the top-level workflow to determine how the WEA workflow would likely process the illegitimate alert message. (To view the top-level WEA workflow, see Figure 5 on page 19.) Based on our analysis, the illegitimate alert message would be accepted by the IPAWS-OPEN Gateway and forwarded to other FEMA systems for processing. Ultimately, the message would make its way to the wireless devices of people who gathered for the event.

6.6 Stakeholder Consequences

Workflow consequences indicate how a threat might affect the objectives of a workflow or mission thread. This is a necessary part of a security risk analysis. However, it is not sufficient. To conduct a thorough security risk analysis, we must look beyond a threat's effect on the workflow or mission thread and examine how the stakeholders of that workflow or mission thread might be affected. Table 6 provides the details of our stakeholder analysis.

[25] The AO operator is not actively monitoring the AOS log because no alert has been issued by the AO. As a result, no one at the AO is aware that an illegitimate alert has been sent.

Table 6: Stakeholder Consequences

Stakeholder	Consequence
Recipients	▪ Some people will ignore the message and take no action. ▪ Some people will believe the message and decide to leave the area. ▪ People could be put in harm's way from the resulting panic, leading to injuries and death.
Alert Originators	▪ Alert originators could be held liable for damages. ▪ The reputations of alert originators could be damaged.
FEMA	▪ The reputation of WEA could be damaged.
CMSPs	▪ The reputation of service providers could be damaged.
Alert Originators/ FEMA/CSMPs	▪ Future attacks could become more likely (i.e., copy-cat attacks).

Stakeholders can experience a variety of risk-relevant consequences, including health, safety, legal, financial, and reputation consequences. Ultimately, we use the stakeholder consequences when we evaluate the impact of a security risk. (We address the basic requirements of evaluating impact in Section 7.3 of this report.)

6.7 Enablers

Enablers are the conditions and circumstances that lead to the occurrence of a risk. Enablers include

- vulnerabilities (i.e., design weaknesses, coding errors, configuration errors) that a threat actor could exploit to produce an adverse consequence or loss
- any additional conditions or circumstances that are needed for the risk to occur

Table 7 highlights the enablers for the AOS spoofing risk. Enabler E3 is an example of a vulnerability (i.e., a design weakness) that the threat actor exploits. Enabler E3 indicates that AO certificates do not have an expiration date or time stamp. Because AO certificates do not expire, a threat actor can use an AO certificate long after it has been obtained. The window of opportunity for any threat that incorporates a stolen IPAWS certificate is not bounded by time.

In contrast, enabler E6 is an example of an enabler that is not designated as a vulnerability. This enabler describes a condition related to the timing of the risk that increases the likelihood that the actor will achieve his or her goal by maximizing the consequences of the risk. Enabler E6 is a condition necessary to realize the risk, but it is not a design weakness, coding error, or configuration error (i.e., not a vulnerability).

Table 7: Enablers

Enabler		Threat Step(s)
E1.	The certificate and encryption key are stored in multiple places (e.g., AOS, AOS database, backup systems, staff computers, offsite backups) within multiple organizations (AO, AOS vendor, FEMA). The actor has many targets for social engineering.	1-4
E2.	If certificate management is inadequate (e.g., poor access controls), then more people could have access to the certificate and encryption key than required.	1-4
E3.	AO certificates do not have an expiration date/time stamp. Because AO certificates do not expire, an actor can use an AO certificate long after it has been obtained.	1-4
E4.	If people at the AO, AOS vendor, or FEMA are susceptible to social engineering, then the threat actor could obtain an electronic copy of the certificate and encryption key.	1-4
E5.	The CAP-compliant message format is documented in reports that are available to the public.	5
E6.	If the actor carefully selects the event to which the illegitimate CAP-compliant message refers, then damages can be maximized.	6
E7.	IPAWS-OPEN does not ask the AOS for confirmation to send a CAP-compliant message through the WEA pipeline. A spoofed message will be forwarded through the WEA pipeline without the knowledge of AO staff.	7-10
E8.	If the AOS does not continuously monitor the IPAWS-OPEN queue, then AO staff might not learn about a spoofed message until stakeholder consequences are observed. The AO will not be able to issue a timely cancellation of the message.	9

Through our piloting activities, we have determined that a security risk scenario and its related data structures provide a useful format for expressing complex security risks. As a result, we have made scenario-based risk analysis the centerpiece of the SERA Framework. In the next section, we introduce the SERA Framework by highlighting its core tasks and steps.

7 SERA Framework Overview

The SERA Framework comprises the following four tasks:

1. Establish operational context.
2. Identify risk.
3. Analyze risk.
4. Develop control plan.

The SERA Framework can be self-applied by the person or group that is responsible for acquiring and developing a software-reliant system or facilitated by external parties on behalf of the responsible person or group.[26] In either case, a small team of approximately three to five people, called the *Analysis Team*, is responsible for implementing the framework and reporting findings to stakeholders.

An Analysis Team is an interdisciplinary team that requires team members with diverse skill sets. Examples of skills and experience that should be considered when forming a team include security engineering risk analysis, systems engineering, software engineering, operational cybersecurity, and physical/facility security. The exact composition of an Analysis Team depends on the point in the lifecycle in which the SERA Framework is being applied and the nature of the engineering activity being pursued. The Analysis Team begins its work by focusing on the environment in which a software-reliant system will be deployed.

7.1 Establish Operational Context (Task 1)

Task 1 defines the operational context for the analysis. First, the Analysis Team identifies the system of interest for the analysis and then determines how the system of interest supports operations (or is projected to support operations if the system of interest is not yet deployed).

Each software application or system typically supports multiple operational workflows or mission threads during operations. The goal is to (1) select which operational workflow or mission thread the team will include in the analysis and (2) document how the system of interest supports the selected workflow or mission thread. This establishes a baseline of operational performance for the system of interest. The team then analyzes security risks in relation to this baseline. Table 8 highlights the three steps performed during this task.

[26] A facilitated assessment still requires participation from groups that are respons ble for acquiring and developing the system of interest. The person facilitating the assessment has expertise in conducting security risk analysis. The facilitator includes others on the team with skills and experience in other areas, such as systems engineering, software engineering, operational cybersecurity, and physical/facility security.

Table 8: Task 1: Steps Performed

Step	Description	Output
1.1 Determine system of interest.	The Analysis Team identifies the system of interest for the analysis. The system of interest is the software application or system that is the focus of the analysis. Selecting the system of interest defines the scope of the subsequent analysis.	System of interest
1.2 Select workflow/ mission thread.	After selecting the system of interest, the Analysis Team determines which workflows or mission threads to include in the analysis. The system of interest might support multiple workflows or mission threads during operations. Selecting relevant workflows or mission threads helps to refine the scope of the analysis further.	Selected workflows/ mission threads
1.3 Establish operational views.	In the final step of Task 1, the Analysis Team establishes a common view of the operational environment in which the system of interest must function. The team uses one or more models to characterize the following operational views: ▪ workflow/mission thread ▪ stakeholder ▪ data ▪ network ▪ physical ▪ use case These views provide team members with the information they need to begin identifying risk scenarios in Task 2.	Operational models

Descriptions of operational views needed to conduct Step 1.3 of the framework are provided in Section 5.1 of this report. As part of their day-to-day job duties, various engineering, organizational, and stakeholder groups will have already developed models consistent with the SERA views. Consider the following examples:

- Engineers should have developed a to-be state for business processes or mission threads that are supporting by the subsystems of interest. This is part of good engineering practice. All relevant workflows or mission threads should be made available to the Analysis Team.
- Software and system engineers should have developed uses cases for the system of interest. All relevant use cases should be made available to the Analysis Team
- In most cases, the system of interest will be placed into an existing operational network (or networks). Current topology diagrams should exist for all relevant networks.
- In most cases, the system of interest will be placed into an existing facility (or facilities). Current diagrams (e.g., office layouts) should exist for all relevant facilities.

Compiling existing models helps to reduce the scope of Task 1. However, our experience indicates that many organizations do not follow good engineering practices; one or more of the required models might not be documented in a suitable format.

7.2 Identify Risk (Task 2)

Task 2 focuses on risk identification. In this task, the Analysis Team transforms a security concern into a distinct, tangible risk scenario that can be described and measured. The team starts by

reviewing the operational models from Task 1. It then identifies the basic threat that is causing concern as well as the sequence of steps required for that threat to be realized. During threat identification, the Analysis Team might refer to a Library of Threat Archetypes for guidance. In this context, we define a *threat archetype* to be a pattern or model that illustrates the key characteristics of a complex threat scenario. The team can then tailor relevant threat archetypes to the given situation. A Library of Threat Archetypes is a collection of threat archetypes that the team considers during Step 2.1 of the framework.[27]

Next, the team estimates how each threat will affect the workflow or mission thread and selected stakeholders. Finally, the Analysis Team creates the narrative for the security risk scenario and compiles all data related to the scenario in a usable format. Table 9 highlights the specific steps performed during this task. *The steps in Table 9 are performed for each risk that is identified*. We present examples of a security risk scenario and associated supporting data in Section 6 of this report.

Table 9: Task 2: Steps Performed

Step		Description	Output
2.1	Identify threat.	The Analysis Team first analyzes the operational models from Task 1 to identify critical data that are transmitted, stored, and processed by the system of interest (i.e., critical assets). The team then examines how threat actors might violate the security attributes (i.e., confidentiality, integrity, availability) of the critical data. For threats that the team will analyze further, it documents the components of the threat and the sequence of steps required to execute the threat (i.e., threat sequence).	Threat components Threat sequence
2.2	Establish consequence.	The next step in the analysis is to establish the consequences of each threat identified during the previous step. In this step, the Analysis Team analyzes the workflow/mission thread and stakeholder models from Task 1 to determine how the workflow/mission thread and stakeholders could be affected by that threat.	Workflow consequences Stakeholder consequences
2.3	Identify enablers.	Enablers include vulnerabilities that a threat actor could exploit as well as the conditions and circumstances that are needed for the risk to occur. In this step, the Analysis Team identifies and documents the enablers of the risk.	Enablers
2.4	Develop risk scenario.	The team documents a narrative description of the security risk based on the information generated in Steps 2.1 through 2.3. Finally, the team documents a risk statement that provides a succinct and unique description of the security risk scenario that is used for tracking purposes.	Risk scenario Risk statement

7.3 Analyze Risk (Task 3)

Task 3 is focused on risk analysis. During this task, the Analysis Team evaluates each risk in relation to predefined criteria to determine its probability, impact, and risk exposure. The steps performed during Task 3 are featured in Table 10.

[27] The L brary of Threat Archetypes is a concept that we are currently developing. In future reports, we intend to provide details about the structure of the library and how it is used during threat identification.

Table 10: Task 3: Steps Performed

Step	Description	Output
3.1 Establish probability.	A risk's probability provides a measure of the likelihood that the risk will occur. In Step 3.1, the Analysis Team determines and documents the probability of occurrence for the security risk scenario.	Probability
3.2 Establish impact.	A risk's impact is a measure of the severity of a risk's consequence if the risk were to occur. The Analysis Team analyzes and documents the impact of the security risk scenario.	Impact
3.3 Determine risk exposure.	Risk exposure is a measure of the magnitude of a risk based on current values of probability and impact. The team determines the risk exposure for the scenario based on the individual values of probability and impact documented in Steps 3.2 and 3.1.	Risk exposure

Our early research and development effort related to implementing the SERA Framework has been directed toward operational modeling (Task 1) and scenario-based expressions of risk (Task 2). In our pilots of the framework to date, we have employed standard, qualitative risk analysis when conducting Task 3. We provide examples of the risk evaluation criteria and results from our WEA pilot in the appendix of this report.

7.4 Develop Control Plan (Task 4)

Task 4 establishes a plan for controlling a selected set of risks. First the Analysis Team prioritizes the security risk scenarios based on their risk measures. Once priorities have been established, the team determines the basic approach for controlling each risk (i.e., accept or plan[28]) based on predefined criteria and current constraints (e.g., resources and funding available for control activities). For each risk that is not accepted, the Analysis Team develops a control plan that indicates

- how the threat can be monitored and the actions taken when it is occurring (recognize and respond)
- which protection measures can be implemented to reduce vulnerability to the threat and minimize any consequences that might occur (resist)
- how to recover from the risk if the consequences or losses are realized (recover)

[28] The SERA Framework examines control approaches in Steps 4.2 and 4.3. During Step 4.2, the Analysis Team determines which risks will be accepted and no longer considered and which will have control plans. At this point in applying the framework, the Analysis Team does not identify specific strategies for transferring, avoiding, and mitigating risks. Those strategies are addressed in Step 4.3. As outlined in Sections 6.4 through 6.7 of this report, security risk scenarios comprise multiple threat steps (as defined in the threat sequence), many enablers, and a range of indirect consequences. An Analysis Team might employ multiple strategies for addressing a given security risk scenario. For example, some steps in the threat sequence might be avoided through restructuring the workflow/mission thread or changing the network architecture. Certain financial consequences might be transferred to third parties by purchasing insurance. The probability of occurrence for some steps in the threat sequence or some types of consequences might be reduced by implementing mitigation controls. Specific control strategies (e.g., transfer, avoid, mitigate) are considered when the control plan is being developed.

A subset of the control actions will have implications for the software (or system) requirements and design. The team must determine which control actions might affect the requirements or design of the system of interest and document them for further analysis. Table 11 highlights the three steps performed during Task 4.

Table 11: Task 4: Steps Performed

Step	Description	Output
4.1 Prioritize risks.	The Analysis Team prioritizes all security risk scenarios based on their impact, probability, and risk exposure measures.	Prioritized risk scenarios
4.2 Select the control approach.	During this step, the team determines how it will handle each risk. If a risk is accepted, its consequences will be tolerated; no proactive action to address the risk will be taken. If the team decides to take action to control a risk, it will develop a control plan for that risk in Step 4.3.	Control approach
4.3 Establish control actions.	The Analysis Team defines and documents a plan for all risks that are being controlled. A control plan establishes a range of actions needed to ▪ recognize and respond to threats ▪ resist the threat and potential consequences ▪ recover from consequences when they occur A subset of the control actions will have implications for the software (or system) requirements and design. Any control actions with requirements or design implications are documented for further analysis.	Control plan Candidate design controls

We provide an example of a control plan for the AOS spoofing risk in the appendix of this report. Based on our analysis of the AOS spoofing risk, we identified the following design issues and controls:

- The AOS does not continuously monitor the IPAWS-OPEN queue. A new AOS system capability is needed to enable monitoring of the queue. If a spoofed WEA alert is sent by a threat actor, the AO operator will be notified automatically by the AOS. He or she will be able to send a cancellation message immediately to the IPAWS-OPEN Gateway. This control will not prevent the illegitimate alert from being sent but will provide a timely cancellation message to recipients to prevent them from taking unnecessary actions.
- An AO certificate does not have an expiration date/time stamp. A new WEA requirement is needed to add date/time stamps to all WEA certificates. This increases the probability that the IPAWS certificate will be expired when the threat actor tries to use it. This control will prevent the spoofed message from being processed by the IPAWS-OPEN Gateway. However, this control affects the interface between the AOS and the IPAWS-OPEN Gateway, which is a legacy system. Since the IPAWS-OPEN Gateway is a legacy system, it might not be feasible to make changes to the format of the IPAWS certificate.
- The IPAWS-OPEN Gateway does not ask the AOS for confirmation before sending the CAP-compliant message through the WEA pipeline. A new system capability is needed to enable confirmation. The AO staff would not approve the sending of a message that did not originate within the AO organization. This control will prevent the IPAWS-OPEN Gateway from forwarding a spoofed message. This control also affects the interface between the AOS and the

IPAWS-OPEN Gateway; because of this, it might not be feasible to make changes to the interface.

The costs, benefits, and feasibility of all three controls must be evaluated before determining which control should be included in the AOS system requirements.

Our early piloting of the SERA Framework has produced promising results. The framework provides a foundation for refining and extending our research into early lifecycle security risk analysis. In the next, and final, section of this report, we summarize our work to date, highlight our preliminary pilot results, and present our future plans for this work.

8 Summary and Next Steps

The SERA Framework defines an approach for analyzing security risk in software-reliant systems and systems of systems across the software lifecycle. Traditional security-risk analysis methods are based on a simplified view of security risk, where a single threat actor exploits a single vulnerability in a single system to cause an adverse consequence. However, in reality, multiple actors exploit multiple vulnerabilities in multiple systems as part of a complex chain of events.

The SERA Framework is designed to address complex security risks arising from the network of people, processes, and technologies that form the foundation of today's operational environments. The framework integrates system and software engineering with operational security by requiring engineers to consider operational security risks early in the lifecycle. The SERA Framework is not a quick check-the-box analysis activity. In contrast to many checklist-based approaches for software security, the framework defines an engineering practice for analyzing security risk early in the lifecycle. It requires a detailed examination of complex security issues with the goal of reducing the residual security risk in deployed software-reliant systems.

8.1 Early Piloting of SERA

We started developing the SERA Framework in October 2013. We focused our initial research-and-development activities on the subset of design weaknesses related to missing or incomplete security requirements. We have performed limited piloting of the framework to establish feasibility. Several of the examples in this report were taken from our initial set of pilots.

The initial results from applying the SERA Framework are promising. In each pilot, we assess multiple security risks for the system being developed. We then develop a set of candidate security requirements to address the design weaknesses that are contributing to the high-priority security risks. Next, we compare our candidate security requirements to those that are described in the program's requirements documents. In our limited piloting to date, we have identified several instances of missing or incomplete security requirements. While the results of our pilots are encouraging, we recognize that they have been limited in number and scope. More piloting is needed to more fully characterize the benefits and limitations of our approach.

8.2 Future SERA Development

To date, we have focused on developing the SERA Framework and piloting it with a selected set of early adopters. The framework provides a set of guidelines defining *what* tasks and steps need to be performed when analyzing security risk early in the lifecycle. It does not define *how* to perform those tasks and steps. Our overarching development goals are to (1) define a method that specifies how to conduct the tasks and steps specified in the SERA Framework and (2) transition that method to the systems and software engineering community. While we have made considerable progress, we have a long way to go before we are able to transition the SERA Method to the community.

To achieve our development goals for SERA, we need to address the following activities:

- *Pilot and refine the SERA Framework*. Piloting activities need to include multiple types of projects and programs, ranging from traditional, large-scale DoD development programs to projects applying Agile principles. These piloting activities will help us to refine the framework and ensure its applicability across a wide range of program types and lifecycle models.
- *Develop, refine, and codify a SERA Method that is consistent with the framework*. In our pilots to date, we have developed a prototype set of guidelines, procedures, and artifacts that enable us to perform the tasks and steps specified in the SERA Framework. These guidelines, procedures, and artifacts are the first step toward the formal definition of the SERA Method. As we continue to pilot the framework, we will refine the associated guidelines, procedures, and artifacts. This activity will require considerable engagement with projects and programs throughout the community. Over time, we hope to converge on a standard method for applying the SERA Framework. However, it is possible that we might need to develop multiple variants of the method for different contexts (e.g., based on different types of programs and lifecycle models). Our piloting activities will ultimately determine if we need to develop one or multiple methods.
- *Transition the SERA Framework and Method to the community*. The goal of transition is to enable people throughout the community to perform the SERA Method. This activity requires packaging SERA guidelines, procedures, and artifacts in a way that is easily consumed by people throughout the system and software engineering community. Transitioning a method to the community requires developing support materials, such as reports, books, and courses. Developing these support materials can require considerable resources and time.

We are currently focusing on piloting and refining the SERA Framework. As we pilot the framework, we will continue refining the guidelines, procedures, and artifacts we use to conduct the tasks and steps specified in the framework. Ultimately, the extent to which we are able to codify and transition the SERA Method is predicated on the community's interest in early lifecycle risk analysis. If the community's interest in early lifecycle risk analysis continues to grow, then a market for the SERA Method likely will begin to materialize. This community interest will provide the resources and pilots needed to support our development and transition of the SERA Method. One key to fostering community interest in early lifecycle security risk analysis is the emergence of standards, laws, and regulations that mandate a risk-based approach for software assurance.

8.3 Aligning with Standards, Laws, and Regulations

In the past few years, the notion of using a risk-based approach for software assurance has been gaining momentum within the community. For example, NIST 800-37,[29] which was issued in 2010, provides guidance for applying the NIST Risk Management Framework (RMF) to Federal information systems. The NIST RMF provides a disciplined, structured process for integrating cybersecurity risk management activities into the system development lifecycle. A foundational premise of the RMF is the notion that effective security risk mitigation across all phases of the system development lifecycle is critical to minimizing operational cybersecurity risk. In 2014, the

[29] NIST Special Publication 800-37, Revision 1, is entitled *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach* [NIST 2010].

DoD issued DoD Instruction 8510.01,[30] which establishes the NIST RMF as a replacement for the DoD Information Assurance Certification and Accreditation Process (DIACAP).[31]

DoD Instruction 8510.01 and NIST 800-37 require DoD and Federal acquisition programs to integrate cybersecurity risk management activities into the system development activities across the lifecycle. Moving forward, a key aspect of our research is ensuring that the SERA Framework and Method are consistent with DoD Instruction 8510.01 and NIST 800-37. By doing this, we will provide a codified engineering practice for analyzing and mitigating cybersecurity risk early in the software lifecycle (i.e., requirements, architecture, design) that is consistent with DoD and Federal requirements.

8.4 Final Thoughts

Software assurance provides stakeholders with a level of confidence that (1) a software-reliant system will function as intended when deployed and (2) its cybersecurity risk will be kept within an acceptable tolerance over time. Software is a growing component of business- and mission-critical systems across all government and industry sectors. As software exerts more control of complex systems, the degree of cybersecurity risk in those systems will increase in kind. The community needs better means of controlling cybersecurity risk in deployed systems.

Early lifecycle security risk analysis is an approach that programs can use to reduce residual security risk in deployed software-reliant systems. Analyzing software security risk early throughout the lifecycle helps build confidence that deployed systems will function as intended during operations, even in the event of cybersecurity attacks. In addition, addressing security weaknesses early in the lifecycle rather than waiting until operations should help control the costs associated with operating and maintaining software-reliant systems.

This report describes the SERA Framework, the culmination of our initial research project related to early lifecycle security risk analysis. The initial results from applying the SERA Framework are promising. However, we have many additional avenues to explore. We have highlighted a few of these avenues in this section. Overall, we view the work documented in this report as a starting point for SERA, not as a completed body of work.

---

[30] DoD Instruction 8510.01 is entitled *Risk Management Framework (RMF) for DoD Information Technology (IT)* [DoD 2014].

[31] Prior to March 2014, the DIACAP was the DoD process for ensuring that risk management is applied to DoD information systems. It defined a set of activities for information system certification and accreditation (C&A), with the goal of maintaining the system's information assurance posture over time. In March 2014, the DIACAP was replaced by the NIST RMF.

Appendix: Example Results for Risk Analysis and Control

This appendix provides example results for Tasks 3 and 4 of the SERA Framework.

A.1 Analyze Risk (Task 3)

In our pilots of the framework to date, we have employed qualitative risk analysis when conducting Task 3. In this section of the appendix, we provide the qualitative risk evaluation criteria we used to evaluate WEA risks as well as the risk measures for the AOS spoofing risk.

A.1.1 Probability Evaluation Criteria

Probability is a measure of the likelihood that a risk will occur. Qualitative risk analysis requires people to estimate a risk's probability in relation to a set of predefined criteria. Table 12 provides the criteria that we used to evaluate probability for the WEA pilot that we conducted.

Table 12: Risk Probability Criteria

Value	**Definition**	**Guidelines/Context/Examples** *How often would an event occur for each value? How many times in a given year?*
Frequent (5)	The scenario occurs on numerous occasions or in quick succession. It tends to occur quite often or at close intervals.	≥ one time per month (≥ 12 / year)
Likely (4)	The scenario occurs on multiple occasions. It tends to occur reasonably often, but not in quick succession or at close intervals.	
Occasional (3)	The scenario occurs from time to time. It tends to occur "once in a while."	~ one time per 6 months (~ 2 / year)
Remote (2)	The scenario can occur, but it is not likely to occur. It has "an outside chance" of occurring.	
Rare (1)	The scenario occurs infrequently and is considered to be uncommon or unusual. It is not frequently experienced.	≤ one time every 3 years (≤ .33 / year)

A.1.2 Impact Evaluation Criteria

Impact provides a measure of the severity of a risk's consequence if the risk were to occur. Similar to probability analysis, we develop a set of predefined criteria when analyzing the impact of a risk's consequences. Table 13 illustrates the criteria that we developed for impact analysis.

Table 13: Risk Impact Criteria

Value	Definition
Maximum (5)	The impact on the organization is severe. Damages are extreme in nature. Mission failure has occurred. Stakeholders will lose confidence in the organization and its leadership. The organization either will not be able to recover from the situation, or recovery will require an extremely large investment of capital and resources. Either way, the future viability of the organizational is in doubt.
High (4)	The impact on the organization is large. Significant problems and disruptions are experienced by the organization. As a result, the organization will not be able to achieve its current mission without a major re-planning effort. Stakeholders will lose some degree of confidence in the organization and its leadership. The organization will need to reach out to stakeholders aggressively to rebuild confidence. The organization should be able to recover from the situation in the long run. Recovery will require a significant investment of organizational capital and resources.
Medium (3)	The impact on the organization is moderate. Several problems and disruptions are experienced by the organization. As a result, the organization will not be able to achieve its current mission without some adjustments to its plans. The organization will need to work with stakeholders to ensure their continued support. Over time, the organization will be able to recover from the situation. Recovery will require a moderate investment of organizational capital and resources.
Low (2)	The impact on the organization is relatively small, but noticeable. Minor problems and disruptions are experienced by the organization. The organization will be able to recover from the situation and meet its mission. Recovery will require a small investment of organizational capital and resources.
Minimal (1)	The impact on the organization is negligible. Any damages can be accepted by the organization without affecting operations or the mission being pursued. No stakeholders will be affected. Any costs incurred by the organization will be incidental.

A.1.3 Risk Exposure Matrix

Risk exposure is a measure of the magnitude of a risk based on current values of probability and impact. The matrix used to evaluate a risk exposure is shown in Table *14*.

Table 14: *Risk Exposure Matrix*

Impact	Probability: Rare (1)	Remote (2)	Occasional (3)	Probable (4)	Frequent (5)
Maximum (5)	Medium (3)	Medium (3)	High (4)	Maximum (5)	Maximum (5)
High (4)	Low (2)	Low (2)	Medium (3)	High (4)	Maximum (5)
Medium (3)	Minimal (1)	Low (2)	Low (2)	Medium (3)	High (4)
Low (2)	Minimal (1)	Minimal (1)	Minimal (1)	Low (2)	Medium (3)
Minimal (1)	Minimal (1)	Minimal (1)	Minimal (1)	Minimal (1)	Low (2)

A.1.4 Risk Measures

Table 15 provides the risk measures for the AOS spoofing risk. We determined the measures using the three sets of criteria defined above. In the table, we also provide the rationale underlying our estimates of probability and impact. Qualitative risk analysis uses a "mechanical" process for determining risk exposure. For the AOS spoofing risk, we determined the impact to be between *maximum* and the probability to be *rare*. The intersection between the probability and impact values in the risk exposure matrix (Table 14) yields a risk exposure of *medium*.

Table 15: Risk Measures

Probability		**Impact**		**Risk Exposure**	
Measure	Rare	**Measure**	Maximum	**Measure**	Medium
Rationale	▪ This risk requires that a complex sequence of events occurs. ▪ The actor has to be highly motivated. ▪ The attack needs to coincide with an event where a large crowd will gather.	**Rationale**	▪ The impact will ultimately depend on whether people trust the WEA service and take the action recommended in the illegitimate WEA alert. ▪ Health and safety damages could be severe, leading to potentially large legal liabilities. ▪ The reputation of WEA could be severely damaged beyond repair.		

A.2 Develop Control Plan (Task 4)

Task 4 ultimately produces a plan for controlling selected risks. First, risks are prioritized and put into a spreadsheet format. Once priorities have been established, the team determines an approach for addressing each risk (i.e., accept or control). For each risk that will be controlled, a control plan is then developed. In this section of the appendix, we present (1) a prioritized risk spreadsheet that includes four WEA risks and (2) candidate control actions for the AOS spoofing risk.

A.2.1 Prioritized Risk Spreadsheet

We used the following guidelines before prioritizing the list of WEA risks:

- Impact was the primary factor for prioritizing security risks. Risks with the largest impacts are deemed to be of highest priority.
- Probability was the secondary factor for prioritizing security risks. Probability is used to prioritize risks that have equal impacts. Risks of equal impact with the largest probabilities are considered to be the highest priority risks.

The prioritized risk spreadsheet for our WEA pilot is shown in Table 16. In the table, we also included the control approach that we selected for each risk.

Table 16: Prioritized Risk Spreadsheet

ID	Risk Statement	Impact	Probability	Risk Exposure	Approach
1	**IF** an outside actor with malicious intent obtains a valid certificate through social engineering and uses it to send an illegitimate CAP-compliant message by spoofing an AOS, **THEN** health, safety, legal, financial, and reputation consequences could result.	Maximum	Rare	Medium	Plan
3	**IF** an insider with malicious intent spoofs the identity of a colleague and sends an illegitimate CAP-compliant message, **THEN** public trust in the WEA service could erode and organizational reputation consequences could result.	Med	Rare-Re-mote	Min-Low	Plan
2	**IF** malicious code prevents an operator from entering an alert into the AOS, **THEN** health, safety, legal, financial, and productivity consequences could result.	Low-Med	Remote	Min-Low	Plan
4	**IF** the internet communication channel for the AOS is unavailable due to a cybersecurity attack on the ISP, **THEN** health and safety consequences could result.	Low-Med	Remote	Min-Low	Plan

A.2.2 Candidate Control Actions

A control plan defines a set of actions for addressing a risk. These plans include actions from the following categories:

- *Recognize and respond*—Monitor the threat and take action when it is detected.
- *Resist*—Implement protection measures to reduce vulnerability to the threat and minimize any consequences that might occur.
- *Recover*—Recover from the risk if the consequences or losses are realized.

Table 17 provides the candidate control actions that we developed for the WEA spoofing risk. Each action is linked to one or more enablers that we identified for the risk. (See Table 7 in Section 6.7 for the complete list of enablers for the WEA spoofing risk.)

Table 17: Candidate Control Actions

Category	Action	Enabler
Recognize and Respond	The AOS should continuously monitor the IPAWS-OPEN queue. When an IPAWS-OPEN status notification does not match a message sent by the AOS, the AO operator should send a cancellation message to IPAWS-OPEN.	E8
Resist	The AOS should limit access to the AO certificate and encryption key based on AO and vendor roles (i.e., implement role-based access for AO certificates).	E1, E2
	The AO should implement procedures for transmitting and storing the AO certificate and encryption key within AO and vendor systems.	E1, E2
	An AO certificate should have an expiration date/time stamp.	E3

Category	Action	Enabler
	The AO and vendor should provide security awareness training for all employees with access to the AO certificate and encryption key. The topic of social engineering should be covered by the security awareness training.	E4
	The IPAWS-OPEN Gateway should ask the AOS for confirmation before sending the CAP-compliant message through the WEA pipeline.	E8
Recover	The AO should notify FEMA and cancel the compromised AO certificate and encryption key to prevent it from being used again.	E3

The following candidate actions from the above plan have design implications:

- The AOS should continuously monitor the IPAWS-OPEN queue. When an IPAWS-OPEN status notification does not match a message sent by the AOS, the AO operator should send a cancellation message to IPAWS-OPEN.
- An AO certificate should have an expiration date/time stamp.
- The IPAWS-OPEN Gateway should ask the AOS for confirmation before sending the CAP-compliant message through the WEA pipeline.

The costs, benefits, and feasibility of all three controls must be evaluated before determining which control should be included in the AOS system requirements.

References

URLs are valid as of the publication date of this document.

[Alberts 2002]
Alberts, Christopher & Dorofee, Audrey. *Managing Information Security Risks: The OCTAVE*^{SM} *Approach*. Addison-Wesley, 2002. http://www.sei.cmu.edu/library/abstracts/books/0321118863.cfm

[Alberts 2006]
Alberts, Christopher. *Common Elements of Risk* (CMU/SEI-2006-TN-014). Software Engineering Institute, Carnegie Mellon University, 2006. http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=7899

[Bergey 2009]
Bergey, John K. *A Proactive Means for Incorporating a Software Architecture Evaluation in a DoD System Acquisition* (CMU/SEI-2009-TN-004). Software Engineering Institute, Carnegie Mellon University, 2009. http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=8935

[Charette 1990]
Charette, Robert N. *Application Strategies for Risk Analysis*. McGraw-Hill Book Company, 1990.

[DoD 2014]
Department of Defense. *Risk Management Framework (RMF) for DoD Information Technology (IT)* (DoD Instruction 8510.01). Washington, DC, Department of Defense, 2014. http://www.dtic.mil/whs/directives/corres/pdf/851001_2014.pdf

[Dorofee 1996]
Dorofee, A.; Walker, J.; Alberts, C.; Higuera, R.; Murphy, R.; & Williams, R. *Continuous Risk Management Guidebook*. Software Engineering Institute, Carnegie Mellon University, 1996. http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=30856

[Kloman 1990]
Kloman, H. F. "Risk Management Agonists." *Risk Analysis 10,* 2 (June 1990): 201–205.

[Levine 2003]
Levine, Linda; Meyers, B. Craig; Morris, Ed; Place, Patrick R. H.; & Plakosh, Daniel. *Proceedings of the System of Systems Interoperability Workshop (February 2003)* (CMU/SEI-2003-TN-016). Software Engineering Institute, Carnegie Mellon University, 2003. http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=6469

[Maier 1996]
Maier, Mark. "Architecting Principles for Systems-of-Systems." 567-574. *Proceedings of the Sixth Annual International Symposium of INCOSE*. Boston, MA, July 7-11, 1996. INCOSE, 1996.

[Mainstay 2010]
Mainstay Partners. "Does Application Security Pay?" September 2010. http://h30528.www3.hp.com/Security/Fortify_Mainstay_ROI_Study.pdf

[Microsoft 2014]
Microsoft Corporation. "Benefits of the SDL," September 2014. http://www.microsoft.com/security/sdl/about/benefits.aspx

[MITRE 2011]
MITRE Corporation. *2011 CWE/SANS Top 25 Most Dangerous Software Errors*, 2011. http://cwe.mitre.org/top25/

[NASA 2009]
National Aeronautics and Space Administration (NASA). *Final Report, NASA Study on Flight Software Complexity*. NASA Jet Propulsion Laboratory, Systems and Software Division, Pasadena, CA, 2009. http://www.nasa.gov/pdf/418878main_FSWC_Final_Report.pdf

[NDAA 2013]
One Hundred Twelfth Congress of the United States of America. *National Defense Authorization Act for Fiscal Year 2013*. Washington, DC, 2013. http://www.gpo.gov/fdsys/pkg/BILLS-112hr4310enr/pdf/BILLS-112hr4310enr.pdf

[NIA 2010]
Committee on National Security Systems. *National Information Assurance (IA) Glossary CNSS Instruction* (CNSS Instruction No. 4009). Fort George G. Meade, MD, 2010. http://www.dtic.mil/whs/directives/corres/pdf/851001_2014.pdf

[NIST 2010]
National Institute of Standards and Technology. *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach* (NIST Special Publication 800-37 Revision 1). Gaithersburg, MD, National Institute of Standards and Technology, 2014. http://csrc.nist.gov/publications/nistpubs/800-37-rev1/sp800-37-rev1-final.pdf

[NIST 2012]
National Institute of Standards and Technology. *Guide for Conducting Risk Assessments* (NIST Special Publication 800-30 Revision 1). Gaithersburg, MD, National Institute of Standards and Technology, 2012. http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf

[NIST 2014a]
National Institute of Standards and Technology. *Systems Security Engineering: An Integrated Approach to Building Trustworthy Resilient Systems* (NIST Special Publication 800-160, Initial Public Draft). Gaithersburg, MD, National Institute of Standards and Technology, 2014. http://csrc.nist.gov/publications/drafts/800-160/sp800_160_draft.pdf

[NIST 2014b]
National Institute of Standards and Technology. *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.0*. Gaithersburg, MD, National Institute of Standards and Technology, 2014. http://www.nist.gov/cyberframework/upload/cybersecurity-framework-021214.pdf

[Sharp 01]
Sharp, Alec & McDermott, Patrick. *Workflow Modeling: Tools for Process Improvement and Application Development*. Boston, MA: Artech House, 2001.

[Soo Hoo 2001]
Soo Hoo, K. S.; Sudbury, A. W.; & Jaquith, A. R. "Tangible ROI through Secure Software Engineering." *Secure Business Quarterly 1,* 2 (Fourth Quarter 2001).

[WEA 2014]
WEA Project Team. *Wireless Emergency Alerts (WEA) Cybersecurity Risk Management Strategy for Alert Originators* (CMU/SEI-2013-SR-018). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2014. http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=70071

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	November 2014	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Introduction to the Security Engineering Risk Analysis (SERA) Framework	FA8721-05-C-0003

6. AUTHOR(S)

Christopher Alberts, Carol Woody, & Audrey Dorofee

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2014-TN-025

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFLCMC/PZE/Hanscom Enterprise Acquisition Division 20 Schilling Circle Building 1305 Hanscom AFB, MA 01731-2116	n/a

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

Software is a growing component of modern business- and mission-critical systems. As organizations become more dependent on software, security-related risks to their organizational missions are also increasing. Traditional security-engineering approaches rely on addressing security risks during the operation and maintenance of software-reliant systems. However, the costs required to control security risks increase significantly when organizations wait until systems are deployed to address those risks. It is more cost effective to address software security risks as early in the lifecycle as possible. As a result, researchers from the CERT® Division of the Software Engineering Institute (SEI) have started investigating early lifecycle security risk analysis (i.e., during requirements, architecture, and design). This report introduces the Security Engineering Risk Analysis (SERA) Framework, a model-based approach for analyzing complex security risks in software-reliant systems and systems of systems early in the lifecycle. The framework integrates system and software engineering with operational security by requiring engineers to analyze operational security risks as software-reliant systems are acquired and developed. Initial research activities have focused on specifying security requirements for these systems. This report describes the SERA Framework and provides examples of pilot results.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Security Engineering Risk Analysis Framework, SERA Framework, risk analysis, software-reliant systems, systems of systems, software security	61

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL